

# PERFECT SAMPLING OF GENERALIZED JACKSON NETWORKS

J. BLANCHET AND X. CHEN

**ABSTRACT.** We provide the first perfect sampling algorithm for a Generalized Jackson Network of FIFO queues under arbitrary topology and non-Markovian assumptions on the input of the network. We assume (in addition to stability) that the interarrival and service times of customers have finite moment generating function in a neighborhood of the origin, and the interarrival times have unbounded support.

## 1. INTRODUCTION

We present the first perfect sampling algorithm (i.e. unbiased sampling also known as exact simulation) for the steady-state of so-called Generalized Jackson Networks (GJNs).

A precise description of a GJN consists of  $d$  single server queueing stations, with infinite capacity waiting rooms and each operating under a standard FIFO protocol. The  $i$ -th station receives arrivals from outside the network (i.e. external arrivals) according to a renewal process with arrival rate  $\lambda_i \in [0, \infty)$  (note that  $\lambda_i = 0$  is possible, meaning that the  $i$ -th station does not receive external arrivals, but we assume that  $\lambda_i > 0$  for some  $i \in \{1, \dots, d\}$ ). All the renewal arrival processes are independent. We use  $\lambda = (\lambda_1, \dots, \lambda_d)^T$  to denote the vector of arrival rates. (Throughout this paper all vectors are column vectors unless otherwise stated, and we use  $^T$  to denote transposition.)

All the service requirements are independent. Inter-arrival times and service requirements are all independent. The mean service time at station  $i$  is  $1/\mu_i \in (0, \infty)$ . We use  $\mu = (\mu_1, \dots, \mu_d)^T$  to denote the vector of service rates. The service requirements at station  $i$  are i.i.d. (independent and identically distributed).

Immediately after a customer is served at the  $i$ -th station, he will go to station  $j$  with probability  $Q_{i,j} \in [0, 1]$  for  $j \in \{1, \dots, d\}$  and he will leave the network with probability  $Q_{i,0} = 1 - \sum_{j=1}^d Q_{i,j}$ . We write  $Q = (Q_{i,j} : 1 \leq i, j \leq d)$  for the associated  $d \times d$  sub-stochastic routing matrix. The network is assumed to be open in the sense that  $Q^n \rightarrow 0$  as

---

*Date:* February 16, 2016.

*Key words and phrases.* Perfect sampling, Generalized Jackson Networks, Dominated Coupling From The Past, Renewal Theory.

$n \rightarrow \infty$ . We assume, without the loss of generality, that  $Q_{i,i} = 0$ . Otherwise we can redefine the service requirements via a geometric convolution with success probability equal to  $1 - Q_{i,i}$  and thus represent the network in terms of a model in which  $Q_{i,i} = 0$ .

The so-called flow equations are given by

$$(1) \quad \phi_i = \lambda_i + \sum_{j=1}^d Q_{j,i} \phi_j,$$

which implies that  $\phi = (\phi_1, \dots, \phi_d)^T$  satisfies  $\phi = (I - Q^T)^{-1} \lambda$ . (Note that  $(I - Q)^{-1} = I + Q + Q^2 + \dots$  is well defined because the network is open.)

Under the previous setup, the GJN is stable (in the sense of possessing a steady-state distribution for the workload and queue length processes at each station) if and only if

$$(2) \quad \phi < \mu,$$

where the inequality is understood componentwise.

Under mild assumptions (including for example the case of Poisson arrivals or phase-type inter-arrival and service times) we provide the first exact simulation algorithm for a Generalized Jackson Network. (The precise assumptions, listed as Assumptions 1-4, are given in Section 2.2.) All previous algorithms operate under more-restrictive assumptions relative to what is required in our algorithm. The more restrictive assumptions include: a) The networks are Markovian (i.e. inter-arrivals and service times are assumed to be exponential), or b) The networks are bounded (i.e. the stations are assumed to have rooms with finite buffer sizes); see, for example, [4] and [10].

The work of [1] is closest in spirit to our algorithm here. The authors in [1] consider a so-called stochastic fluid network (SFN), which is much simpler than a GJN because there is much less randomness in the system. Customers that arrive at station  $i$  in a SFN bring service requirements which are i.i.d., this part is common to the GJN model. However, the workload is processed and transmitted to the stations in the network in the form of a fluid; so  $Q_{i,j}$  represents the exact proportion of flow from station  $i$  to  $j$ . Therefore, in particular, in a SFN there is no concept of queue-length. In addition, the SFNs treated in [1] has Poisson or Markov modulated arrivals and so even the arrival processes that we consider here are more general. We extend the algorithm in [1] in order to deal with arbitrary renewal processes (as opposed to only Poisson arrivals), the condition on Assumption 2 is needed to apply the technique of [1] based on a suitable exponential tilting (see also [6] and [2]), this connection to exponential changes of measure explains the need for Assumption 3.

The algorithm in [1] allows to obtain a sample from the maximum from time 0 to infinity, of a multidimensional random walk with negative drift. Here we extend the algorithm to sample from the running maximum (componentwise), that is, the maximum from time  $n$  to infinity, for all  $n \geq 0$ . Our extension is given in Algorithm 4.

The real difficulty in doing perfect sampling of GJNs, however, arises from the fact that each customer might bring an arbitrarily long sequence of service requirements, because the description of the routing topology admits the possibility of visiting a given station multiple times. In addition, contrary to SFN's, GJN's are not monotone in their initial condition. This lack of monotonicity introduces challenges when applying standard perfect simulation techniques.

Our strategy is to apply Dominated Coupling From The Past (DCFTP), which requires the use of a suitable dominating process simulated backwards in time and in stationarity. We are able to use sample path comparison results developed by [5], which allow us to bound the total number of customers in the GJN by a set of suitably defined autonomous queues which are correlated. In addition, we provide additional sample path comparison results which are of independent interest (see Theorem 1).

We need to simulate, backwards in time, stationary and correlated autonomous queues. These processes can be represented, componentwise, in terms of an infinite horizon maximum of the difference of superposition of renewal processes (the difference having negative drift so the infinite horizon maximum is well defined). The fact that the queues are correlated comes from the fact that each jump in the renewal processes may correspond to a departure from one station, and at the same time, an arrival to another station due to the internal routing. We are able to extend the technique in [1] in order to deal with multidimensional and correlated renewal processes and thus complete the application of the DCFTP protocol.

The rest of the paper is organized as follows. In Section 2, we briefly discuss how DCFTP operates and describe the GJN. In Section 3, we construct a class of dominating processes which will be useful for our development. We provide a general overview of our algorithm and the main result of the paper in Section 4. Then we proceed by describing how to implement the subroutines of our algorithm in Section 5 and finish the paper with a numerical experiment in Section 6.

## 2. AN INTRODUCTION TO DCFP AND GJN

**2.1. Elements of Dominated Coupling From The Past.** Let us first provide a general description of DCFTP. Consider a stationary process  $(Y(t) : t \in (-\infty, \infty))$ , we are interested in sampling from  $Y(0)$ . Suppose that the following is available to the simulator:

*DCFTP 1* A pair of stochastic processes  $(Y^-(t) : t \in (-\infty, \infty))$  and  $(Y^+(t) : t \in (-\infty, \infty))$  coupled in such a way that  $Y^-(t) \preceq Y(t) \preceq Y^+(t)$  for all  $t$ , where “ $\preceq$ ” is any partial order.

*DCFTP 2* It is possible to simulate  $\bar{\omega} := (Y^-(t), Y^+(t) : t \in [-T, 0])$  for a (finite almost surely) time  $-T$  in the past such that: a)  $Y^+(-T) = Y^-(-T)$ , and b)  $Y(0)$  can be obtained from the information used to generate  $\bar{\omega}$ .

A time  $-T$  satisfying the conditions in *DCFTP 2* is known as a *coalescence time*.

Generally, at least in the setting of Markov processes, the condition that  $Y^+(-T) = Y^-(-T)$  combined with *DCFTP 1* above indicates that the value of  $Y(-T)$  is known and therefore at least the marginal evolution of  $Y(\cdot)$  is completely determined, and so is the value of  $Y(0)$ . However, it is important to keep in mind that the processes  $Y^+$ ,  $Y^-$  and  $Y$  must remain coupled.

The validity of DCFTP is proved in [9]; the method is an extension of CFTP, which was proposed in the seminal paper of [11]. Intuitively, the idea is that if one could simulate the path  $(Y^-(t), Y(t), Y^+(t) : t \leq 0)$ , from the infinite past, then one could obtain  $Y(0)$  in stationarity. However, since we can simulate  $\bar{\omega}$  in finite time and use this information to reconstruct  $Y(0)$ , we do not need to simulate the process from the infinite past.

Obtaining the elements described in bullets *DCFTP 1-2* above often requires several auxiliary constructions. In our particular application  $Y(\cdot)$  corresponds to the number in system in each station (so  $Y(\cdot)$  is a  $d$ -dimensional process) and we shall set  $Y^-(t) = 0$ . The partial order relationship “ $\preceq$ ” is based on the sum of the coordinates (i.e  $x \preceq y$  if and only if  $\sum x_i \leq \sum y_i$ ).

The process  $Y^+(\cdot)$  is the one that will require auxiliary constructions, we shall first construct an auxiliary process  $Y^0$  which dominates  $Y$  based on artificially increasing (just slightly) the service requirement of all stations in the GJN. Then we will construct  $Y^+$  which is a process similar to a GJN, except that the servers will enjoy vacation periods whenever there is no customer waiting in queue to be served. Finally, we will need an additional process,  $Y'$ , which is corresponding to the autonomous queues and will allow us to identify the coalescence time  $-T$ .

**2.2. Description of the GJN.** In this section, we give detailed description and assumptions of the generalized Jackson network (GJN) we are going to simulate.

We consider a GJN consisting of  $d$  service stations and each station has a single server. In the rest of our paper, we shall denote the GJN by  $\mathcal{N}$ . The basic assumptions of the GJN  $\mathcal{N}$  is as follows: .

- *Arrival times:* Customers arrive (from the external world) at station  $i$  according to some renewal process with i.i.d. interarrival times  $U_i(n)$ . In particular,  $U_i(n) = A_i(n) - A_i(n-1)$  where  $A_i(n)$  is the arrival time of the  $n$ -th customer of station  $i$ . The arrival rate  $\lambda_i$  is defined as  $E[U_i(k)] = 1/\lambda_i \in (0, \infty]$ . If  $\lambda_i = 0$  then  $A_i(n) = \infty$ . (By convention we let  $\lambda_i U_i(1) = 0$  if  $\lambda_i = 0$ .)
- *Service times:*  $\sigma_i(k)$  is the service time of the  $k$ -th customer that is served in station  $i$ .  $\{\sigma_i(k)\}$  is a i.i.d. sequence and independent of the arrival times, routing indicators and service times of the other stations. The service rate  $\mu_i$  is defined as  $E[\sigma_i(k)] = 1/\mu_i$ .
- *Routing mechanism:* After finishing service, the  $k$ -th customer in station  $i$  is assigned with a routing indicator  $r_i(k) \in \{0, 1, 2, \dots, d\}$  and it will leave the network immediately if  $r_i(k) = 0$ , or join the queue of station  $r_i(k)$  otherwise.  $\{r_i(k)\}$  is a i.i.d. sequence and independent of the arrival times, service times and routing indicators of the other stations. The routing probability  $Q_{ij}$  is defined as  $Q_{ij} = P(r_i(k) = j)$ .

Clearly the sequences  $\{A_i(n) : n \geq 1\}$  together with  $\{(r_i(k), \sigma_i(k)) : k \geq 1\}$  for  $i \in \{1, \dots, d\}$  are enough to fully describe the evolution of the queueing network, assuming that the initial state of the network is given. So, let us assume that the network is initially empty and let us write  $Y_i(t)$  to denote the number of customers in the  $i$ -th service station at time  $t$ , including both in the queue and in service, for  $i \in \{1, \dots, d\}$ . As noted in the Introduction, the flow equations are given in equation (1), the vector  $\phi_i$ 's in  $\phi = (\phi_1, \dots, \phi_d)^T$  are called the net-input rates of the GJN.

In addition to the stability condition given in (2), throughout this paper we shall impose the following assumptions:

**Assumptions:**

1. The inter-arrival times have unbounded support. That is, if  $\lambda_i > 0$  then  $P(U_i(1) > m) > 0$  for all  $m \in (0, \infty)$ .
2. There exists  $\delta > 0$  such that for all  $i$

$$(3) \quad \begin{aligned} \sup_{t \geq 0} E[\exp(\delta \lambda_i (U_i(1) - t)) | U_i(1) > t] &< \infty, \\ \sup_{t \geq 0} E[\exp(\delta (\sigma_i(1) - t)) | \sigma_i(1) > t] &< \infty. \end{aligned}$$

In particular,  $\lambda_i U_i(1)$  and  $\sigma_i(1)$  have a finite moment generating function for all  $i$ .

3. The inter-arrival times and service times can be individually simulated exactly, and moreover, we can simulate from exponentially tiltings (i.e. the natural exponential family) associated to these distributions – see equation (10).

4. The inter-arrival times and service times have a continuous distribution.

Assumptions 1 to 4 are relatively mild and encompass a large class of models of interest including Poisson arrivals and phase-type service time distributions (and mixtures thereof). We shall also discuss immediate extensions to the case of Markov modulated GJNs. Assumption 1 ensure that the network will empty infinitely often with probability one. We require the existence of a finite moment generating function because we will apply an extension of a technique developed in [1], which is based on exponential tiltings and importance sampling, therefore the need for Assumption 3. We need the uniformity on exponential moments for conditional excess distributions in Assumption 2 because we apply a Lyapunov bound similar to that developed by [7]. However, we believe that this uniformity requirement is a technical condition and that our main result holds assuming only that (3) is satisfied for  $t = 0$ . Finally, Assumption 4 is introduced for simplicity to avoid dealing with simultaneous events.

Under Assumptions 1 to 4 we provide an algorithm for sampling from the steady-state queue-length and workload processes at each station in the network. The number of random variables required to terminate our proposed procedure has a finite moment generating function in a neighborhood of the origin (in particular the expected termination time of the algorithm is finite).

### 3. CONSTRUCTION OF THE AUXILIARY AND DOMINATING PROCESSES

In this section, we shall construct two dominating processes for  $Y(t)$ , related to vacation queues and autonomous queues.

**3.1. An Auxiliary GJN.** Before constructing the two bounding systems, we need to construct an auxiliary upper bound GJN, which we shall denote by  $\mathcal{N}^0$ . The auxiliary GJN  $\mathcal{N}^0$ , is obtained from the original GJN,  $\mathcal{N}$ , by slightly decreasing the service rates at each station while keeping the network stable. In particular, we shall select constants  $a_i \geq 1$  for  $i \in \{1, \dots, d\}$  momentarily. We define  $\sigma_i^0(k) = \sigma_i(k) a_i$ , and correspondingly set  $\mu_i^0 = \mu_i / a_i$  for  $a_i \geq 1$  so that  $\mu^0 = (\mu_1^0, \dots, \mu_d^0)^T$ , satisfies,

$$(4) \quad \lambda < (I - Q^T) \mu^0,$$

componentwise. It is always possible to pick  $a_i \geq 1$  satisfying (4). In order to see this, reason as follows. First, define  $\mu^0 = (I - Q^T)^{-1}(\lambda + \delta e)$  (where  $e$  is the vector of ones and  $\delta > 0$  is to be chosen). Since  $\phi = (I - Q^T)^{-1} \lambda < \mu$  and the matrix  $(I - Q^T)^{-1}$  has non-negative elements, we can choose  $\delta > 0$  small enough so that  $\mu_i > \mu_i^0$  and therefore

$a_i = \mu_i / \mu_i^0 > 1$ . Moreover, by definition

$$(I - Q^T) \mu^0 = \lambda + \delta e > \lambda.$$

The evolution of  $\mathcal{N}^0$ , initially empty, is also fully described by the sequences  $\{A_i(n) : n \geq 1\}$  and  $\{(r_i(k), \sigma_i^0(k)) : k \geq 1\}$ ,  $i \in \{1, \dots, d\}$ , where  $\sigma_i^0(k) = a_i \sigma_i(k)$ . Let  $Y_i^0(t)$  be the number of customers in the  $i$ -th service station at time  $t$  (including both in queue and in service), for  $i \in \{1, \dots, d\}$ . As we shall review in Theorem 1, given the same initial condition at time 0,  $\sum_i Y_i^0(t) \geq \sum_i Y_i(t)$ , for all  $t$ ; this is intuitive since every customer in  $\mathcal{N}^0$  needs more service time at every station than in  $\mathcal{N}$ .

**3.2. The Vacation System.** We now describe the bonding system consisting of vacation queues, which we shall denote by  $\mathcal{N}^+$ . The system  $\mathcal{N}^+$  evolves following almost the same rules as  $\mathcal{N}^0$  except that, whenever the  $i$ -th server completes a service *and* no customer is waiting in queue to be served, the server enters a vacation period following the same distribution of  $\sigma_i^0(k)$ . The vacation periods are all independent, and also independent of the arrival times, service times and routing indicators. If at least one customer is waiting in queue, the server will work on the service requirement of the first customer waiting in queue.

In more detail, the vacation periods are not interrupted when a new customer arrives, instead the customer waits until the server finishes its current activity (current vacation or service). Moreover, if after completing a vacation the server still finds the queue empty, a new vacation period starts, and the server keeps taking vacation periods until, upon return of a vacation, the server finds at least one customer present in the queue, waiting to be served.

The evolution of the vacation system  $\mathcal{N}^+$ , coupled with  $\mathcal{N}^0$ , is fully described by the sequences  $\{A_i(n) : n \geq 1\}$   $\{(r_i(k), \sigma_i^0(k)) : k \geq 1\}$ ,  $i \in \{1, \dots, d\}$ , along with the vacation period sequence  $\{v_i^0(k) : k \geq 1\}$ . For each  $i$ , the sequence  $\{v_i^0(k) : k \geq 1\}$  is an i.i.d. copy of the sequence  $\{\sigma_i^0(k) : k \geq 1\}$ . The random variable  $v_i^0(k)$  denotes the  $k$ -th vacation period taken by the  $i$ -th server.

Let us write  $Y_i^+(t)$  to denote the number of customers in the  $i$ -th station at time  $t$  (including both in queue and in service). As stated in Theorem 1 below, we have that, given the same initial condition at time 0,  $\sum_i Y_i^+(t) \geq \sum_i Y_i^0(t)$  for all  $t$ ; this is intuitive since every customer in  $\mathcal{N}^+$  keeps the same service time and routing (relative to  $\mathcal{N}^0$ ), but the departure times must occur later due to the vacation periods.

**3.3. The Autonomous System.** The final bounding system is a set of the so-called autonomous queues which we shall denote by  $\mathcal{N}'$ . In this subsection, we shall describe the

evolution of this system and provide an expression for its number of customers in queue. In the next subsection, we shall explain how  $\mathcal{N}'$  is coupled with  $\mathcal{N}^+$ .

Define  $(N_i(t) : t \geq 0)$  to be the non-delayed renewal process corresponding to the sequence  $\{A_i(n) : n \geq 1\}$ ; that is, defining  $A_i(0) = 0$ , by convention we have

$$N_i(t) = \max\{n \geq 0 : A_i(n) \leq t\}.$$

Of course,  $N_i(t) \equiv 0$  if  $\lambda_i = 0$ .

We let  $(V_i^0(k) : k \geq 1)$  be a sequence of i.i.d. random variables with the same distribution  $\sigma_i^0(k)$  (and therefore as  $v_i^0(k)$ ). We write  $B_i(0) = 0$  and set  $B_i(n) = V_i^0(1) + \dots + V_i^0(n)$ . Then, define a renewal process

$$D_i(t) = \max\{n \geq 0 : B_i(n) \leq t\}.$$

Moreover, for each  $i \in \{1, \dots, d\}$  we define a sequence of i.i.d. random variables  $(r'_i(k) : k \geq 1)$  such that

$$P(r'_i(k) = j) = Q_{i,j},$$

for all  $j \in \{0, 1, \dots, d\}$ . We then define

$$D_{i,j}(t) = \sum_{k=1}^{D_i(t)} I(r'_i(k) = j) \text{ (so that } D_i = \sum_{j=0}^d D_{i,j} \text{)}.$$

The random variables  $V_i^0(k)$ 's and  $r'_i(k)$ 's are all mutually independent and independent of the  $A_i(k)$ 's for all  $i \in \{1, \dots, d\}$  and  $k \geq 1$ .

Let  $Y'_i(t)$  be the number of customers in the queue at the  $i$ -th station of  $\mathcal{N}'$ . By the definition of autonomous queues,  $Y'_i(\cdot)$  evolves according to the following Stochastic Differential Equation

$$(5) \quad dY'_i(t) = dN_i(t) + \sum_{j:j \neq i, 1 \leq j \leq d} dD_{j,i}(t) - I(Y'_i(t_-) > 0) dD_i(t),$$

$$Y'_i(0) = 0.$$

In simple words, the number of customers in queue at the  $i$ -th station increases when there is an external arrival ( $dN_i(t) = 1$ ) or an arrival (either virtual or true, see the explanation in Section 3.4) from any other station ( $\sum_{j=0}^d dD_{j,i}(t) = 1$ ), and it decreases at time  $t$  after the completion of an activity (service or vacation, see the explanation in Section 3.4) only if the queue is not empty (i.e.  $I(Y'_i(t_-) > 0)$  and  $dD_i(t) = 1$ ).



One nice property of  $\mathcal{N}'$  is that we have a convenient expression for  $Y_i'(t)$ , which is essential for our CFTP algorithm to work. Let's define

$$X_i(t) = N_i(t) + \sum_{j:j \neq i, 1 \leq j \leq d} D_{j,i}(t) - D_i(t),$$

recall that  $Q_{i,i} = 0$  so we have that  $D_{i,i}(t) = 0$ , and thus we also can write  $\sum_{j=1}^d D_{j,i}(t)$  in the previous display. Then, one can verify that the (unique) solution to equation (5) is given by (see for instance, [8])

$$Y_i'(t) = X_i(t) - \min_{0 \leq s \leq t} X_i(s) = \max_{0 \leq s \leq t} (X_i(t) - X_i(s)).$$

**3.4. Coupling between  $\mathcal{N}'$  and  $\mathcal{N}^+$ .** In order to describe the coupling between  $\mathcal{N}'$  and  $\mathcal{N}^+$ , let us provide an interpretation of the SDE (5) describing  $\mathcal{N}'$ . The evolution of the  $i$ -th queue in  $\mathcal{N}'$  can be seen as a single server queue with vacation periods. Customers arrive according to the superposition of the processes  $N_i$  and  $(D_{j,i} : 1 \leq j \leq d)$ , the server takes a vacation whenever the queue is empty with a distribution which is identical to that of a generic service time. Arriving customers who find the queue empty must wait to be served only until the current vacation epoch finishes.

The difference between  $\mathcal{N}'$  and  $\mathcal{N}^+$  is that in  $\mathcal{N}^+$  no customers are “transferred” from station  $i$  to  $j$  at the end of a vacation epoch of server  $i$ . Note that these types of transfers actually might occur in  $\mathcal{N}'$  because it could be the case, for instance, that  $Y_i'(t-) = 0$ ,  $dD_i(t) = 1$  and the corresponding  $r_i'(k) = j$  so that  $dD_{i,j}(t) = 1$  and a new customer joins the queue at station  $j$ . Consequently, in  $\mathcal{N}'$  there are two types of customers: a) true customers, as those in  $\mathcal{N}^+$ , which are the ones that correspond to external arrivals (i.e. arrivals from the processes  $N_i$  for  $i \in \{1, \dots, d\}$ ), and their corresponding routes through the network, and b) virtual customers, which does not exist in  $\mathcal{N}^+$ , are the ones generated by empty stations that transfer customers to other stations by the mechanism just described above. Therefore, to couple  $\mathcal{N}'$  and  $\mathcal{N}^+$ , we essentially need to distinguish between the true and virtual customers in  $\mathcal{N}'$ .

Recall that the evolution of  $\mathcal{N}'$  is fully described by the process  $N_i(\cdot)$ ,  $D_i(\cdot)$  and  $D_{ij}(\cdot)$ , and  $\mathcal{N}^+$  by the sequences  $\{A_i(n)\}$ ,  $\{r_i(k), \sigma_i^0(k)\}$  and  $\{v_i(k)\}$ . To describe the coupling of  $\mathcal{N}'$  and  $\mathcal{N}^+$ , we shall explain how to couple the pair of sequences. Roughly speaking, the two systems will share the same external arrivals, and each  $V_i^0(k)$  (recall that  $\{V_i^0(k)\}$  are the inter-renewal times of  $D_i(\cdot)$ ) corresponds to a service time  $\sigma_i(k')$  when a customer is in service and to a vacation period  $v_i(k')$  otherwise. We provide the details next.

In our algorithm, we shall first simulate  $\mathcal{N}'$  on some finite time interval  $[T_1, T_2]$ , the corresponding processes  $N_i(t)$ ,  $D_i(t)$  and  $D_{ij}(t)$  on it, and sequences  $\{A_i(n)\}$  and  $\{(V_i(k), r_i'(k))\}$ .

Then, the number of customers  $Y_i^+(t)$  of the coupled vacation system  $\mathcal{N}^+$  evolves according to the following SDE:

$$\begin{aligned} d\hat{Y}_i^+(t) &= dN_{0,i}(t) + \sum_{j:j \neq i, 1 \leq j \leq d} I(S_j^+(t_-) > 0) dD_{j,i}(t) - I(\hat{Y}_i^+(t_-) > 0) dD_i(t), \\ (6) \quad d\hat{S}_i^+(t) &= (I(\hat{Y}_i^+(t_-) > 0) - I(S_i^+(t_-) > 0)) dD_i(t), \\ Y_i^+(t) &= \hat{Y}_i^+(t) + S_i^+(t). \end{aligned}$$

Here  $S_i^+(t) \in \{0, 1\}$  is the number of customer in service at station  $i$  at time  $t$ . In particular, we shall choose a special initial condition for  $\mathcal{N}^+$  according to the comparison results that we shall explain in Section 3.5:

$$(7) \quad \hat{Y}_i^+(T_1) = Y_i'(T_1), \quad S_i^+(T_1) = 1.$$

The remaining service time of the customer at station  $i$  is the residual jump time of  $D_i(\cdot)$ , i.e.,  $= B_i(D_i(T_1) + 1) - T_1$ . Then, the sequences of  $(\sigma_i^0(n), r_i(n))_{n \geq 1}$  can be extracted as follows.

**Procedure 0: Coupling of  $\mathcal{N}'$  and  $\mathcal{N}^+$ :**

- 1) Input  $N_i(t)$ ,  $D_i(t)$  and  $D_{i,j}(t)$  for  $1 \leq i, j \leq d$  and  $t \in [T_1, T_2]$ . Set  $t_i = T_1$ ,  $k_i^s = 0$ ,  $k_i^v = 0$ , and  $n_i = 1$ .
- 2) Compute  $Y^+(t)$  and  $S^+(t)$  according to (6) and the initial condition (7).
- 3) For each  $i$ , while  $t_i < T_2$ , repeat the following:
  - $t_i \leftarrow t_i + V_i^0(n)$ ;
  - If  $S_i^+(t_i-) = 1$ , update  $k_i^s \leftarrow k_i^s + 1$  and set  $\sigma_i^0(k_i^s) = V_i^0(n_i)$  and  $r_i(k_i^s) = r_i'(n_i)$ . Otherwise, update  $k_i^v \leftarrow k_i^v + 1$  and set  $v_i(k_i^v) = V_i^0(n_i)$ .
  - $n_i \leftarrow n_i + 1$ .

**Lemma 1.** *The extracted  $(\sigma_i^0(k), r_i(k))$  form an i.i.d. sequence and independent of the sequence  $\{A_i(n)\}$ .*

*Proof.* This follows from the strong Markov property of the forward recurrence time processes of the renewal processes  $N_i(\cdot)$ ,  $D_i(\cdot)$  and  $D_{ij}(\cdot)$ .  $\square$

**3.5. Comparison Results and Domination.** Now we have a full description of the three systems  $\mathcal{N}^0$ ,  $\mathcal{N}^+$  and  $\mathcal{N}'$  that are coupled with the original GJN  $\mathcal{N}$ , and their corresponding queue length processes. The following theorem gives the comparison results among the four systems, which are essential in our DCFTP algorithm. Its proof is given in the Appendix.

**Theorem 1.** Suppose that the networks  $\mathcal{N}$ ,  $\mathcal{N}^0$ ,  $\mathcal{N}^+$ , and  $\mathcal{N}'$  are all initially empty and are coupled as described through Section 4.1 to 4.4, then the following holds:

i) For any  $t > 0$ ,

$$\sum_{i=1}^d Y_i(t) \leq \sum_{i=1}^d Y_i^0(t) \leq \sum_{i=1}^d Y_i^+(t).$$

ii) Moreover, for any  $t > 0$ , when  $Y_i'(t) = y_i$ , then the service station  $i$  in system  $\mathcal{N}^+$  must satisfy  $Y_i^+(t) \leq y_i + 1$  and  $S_i^+(t) \in \{0, 1\}$ .

iii) The network  $\mathcal{N}^+$ , driven by the SDE (6), is monotone in the initial condition. In other words,  $y_i \in \{0, 1, \dots\}$  and if  $Y^{++}$ ,  $Y^+$ ,  $Y^{+-}$  satisfy the SDEs (6) with initial conditions  $\hat{Y}_i^{++}(0) = y_i + 1$ ,  $\hat{S}_i^{++}(0) = 1$ ;  $\hat{Y}_i^+(0) \leq y_i$ ,  $\hat{S}_i^+(0) \in \{0, 1\}$ , and  $\hat{Y}_i^-(0) = 0 = \hat{S}_i^-(0)$ , then  $Y_i^{++}(t) \geq Y_i^+(t) \geq Y_i^{+-}(t)$  for all  $t \geq 0$ .

In the next section we explain how to use the previous result order to sample from the stationary distribution of  $\mathcal{N}$ , i.e. the joint distribution of customer numbers at each station, the remaining service requirement of the customers in service, and the remaining times to the next external arrivals to each station in steady state.

#### 4. OUR ALGORITHM AND MAIN RESULT

Given the comparison results Theorem 1, we are now ready to given the main procedure of our DCFTP algorithm. In the rest of the paper, for any ergodic stochastic process  $X(\cdot)$ , we shall denote by  $\bar{X}(\cdot)$  its two-sided stationary version.

##### Main Procedure:

- (1) Choose a constant  $C_T > 0$ . Initialize  $T \leftarrow 0$ .
- (2) Simulate the system  $\mathcal{N}'$  in steady state and backwards in time from  $-T$  until  $-T - C_T$ . Obtain the corresponding processes  $\bar{N}_i(\cdot)$ ,  $\bar{D}_i(\cdot)$ ,  $\bar{D}_{ij}(\cdot)$  and  $\bar{Y}'(\cdot)$  from  $-T$  to  $-T - C_T$ . Update  $T \leftarrow T + C_T$ .
- (3) Initialize a vacation system  $\mathcal{N}^{++}$  at time  $-T$  with  $Y_i^{++}(-T) = \bar{Y}'(-T) + 1$ , all servers occupied ( $S_i^+(-T) = 1$ ), and the corresponding remaining service time equals to the time from  $-T$  to the next jump time of process  $\bar{D}_i(\cdot)$ .
- (4) Compute  $(Y_i^{++}(s) : 0 \leq s \leq T)$ , forward in time according to (6) in Section 3.4 and compute the corresponding sequences  $\{A_i(n)\}$ ,  $\{r_i(k), \sigma_i^0(k)\}$  and  $\{v_i^0(k)\}$  according to Procedure 0.
- (5) If there exists  $\tau \in [0, T]$  such that  $Y_i^{++}(\tau) = 0$  for all  $i$ , then we simulate a GJN  $\mathcal{N}$  forward starting from  $\tau < 0$  to time 0 with  $Y_i(\tau) = 0$  for all  $i$  and driven by the sequence  $\{A_i(n)\}$  and  $\{r_i(k), \sigma_i(k)\}$  where each  $\sigma_i(k) = \sigma_i^0(k)/a_i$ . Output  $Y_i(0)$  and terminate.

(6) Otherwise, (if  $Y^+(t) \neq 0$  for  $t \in [0, T]$ ), go back to Step 2.

The above procedure can be validated by the following heuristic. Suppose  $\bar{N}^+$  is the stationary vacation system coupled with  $\bar{N}'$ . Then, according to Part ii) and iii) of Theorem 1, its queue length process  $\bar{Y}_i^+(t) \leq Y_i^{++}(t)$  for all  $i$  and  $t \in [-T, 0]$ . Therefore, we can conclude that  $\bar{Y}_i(\tau) = 0$  for all  $i$  and hence the coupled stationary GJN  $\mathcal{N}$  must be empty at time  $\tau$  by Part i) of Theorem 1. Then, we can recover the value of the stationary process  $\bar{Y}_i(t)$  for  $t \in [\tau, 0]$  and the output  $\bar{Y}_i(0)$  follows the steady-state distribution.

**Theorem 2.** *The state of the network given by the Main Procedure, including  $Y(0)$  and the remaining service times at each station, follow the stationary distribution of the target GJN. Moreover, let  $N$  be the total number of random variables to terminate the Main Procedure, then there is  $\delta > 0$  such that  $E \exp(\delta N) < \infty$ .*

Step 3 through Step 5 in the Main Procedure can be done according to the coupling mechanism described in Section 3.1, 3.2, 3.4, and in particular, Procedure 0. The most difficult part is the execution of Step 2 and we shall explain this in Section 5. The proof, which is given at the end of Algorithm 4, in Section 7.2, mainly constitutes a recapitulation of our development.

## 5. EXECUTION OF STEP 2 IN MAIN PROCEDURE: STATIONARY CONSTRUCTION AND BACKWARD SIMULATION OF $\bar{N}'$

This section is devoted to explain how to execute Step 2 in Main Procedure, that is, to simulate a stationary version of  $Y'$  backwards in time. We shall explain this simulation procedure in three steps. In Section 5.1, we show a stationary version of  $Y'$  can be expressed by a multi-dimensional point process and its maximum. Then, we show the to simulate the point process and its maximum can be reduced to simulating several random walks jointly with their maximum. In the end, in Section 5.3, we explain how to simulate the random walks and their maximum, following the ideas in [1].

**5.1. Express  $Y'$  by Point Processes.** For each  $i$ , we define  $\bar{N}_{0,i}(\cdot)$  as a two-sided, time stationary, renewal point process with inter-arrival time distribution being i.i.d. copies of  $A_i(n+1) - A_i(n)$ . We write  $\{\bar{A}_i(n) : n \in \mathbb{N}_0 \cup (-\mathbb{N})\}$  for the arrival times associated to  $\bar{N}_{0,i}(\cdot)$ , so that  $\bar{A}(-1) < 0 < \bar{A}(0) < \bar{A}(1)$  and define

$$\bar{N}_{0,i}([a, b]) = \sum_n I(\bar{A}(n) \in [a, b]),$$

for any  $a, b \in (-\infty, \infty)$ .

Similarly, we let  $\bar{D}_i(\cdot)$  to be a two-sided, time-stationary version of  $D_i(\cdot)$  and write  $\{\bar{B}_i(n) : n \in \mathbb{N}_0 \cup (-\mathbb{N})\}$  for the arrival times associated to  $\bar{D}_i(\cdot)$  also in increasing order and so that  $\bar{D}_i(-1) < 0 < \bar{D}_i(0) < \bar{D}_i(1)$ . As before,

$$\bar{D}_i([a, b]) = \sum_n I(\bar{B}_i(n) \in [a, b]).$$

Each  $\bar{B}_i(n)$  is attached to a mark  $\bar{r}'_i(n)$  which are i.i.d. copies of the  $r'_i(n)$ 's. All the  $\bar{A}_i(n)$ 's, the  $\bar{D}_i(n)$ 's, and the  $r'_i(n)$ 's are mutually independent. Finally, for any  $a, b \in (-\infty, \infty)$ , define

$$\bar{D}_{i,j}([a, b]) = \sum_n I(\bar{B}_i(n) \in [a, b], \bar{r}'_i(n) = j).$$

Intuitively,  $N_{0,i}(\cdot)$  describes the external arrivals to station  $i$ ,  $D_{i,0}(\cdot)$  describes the potential departures from station  $i$ , and  $D_{ij}(\cdot)$  describes the potential internal routings from station  $i$  to  $j$ . For all  $t \geq 0$ , we define

$$\begin{aligned} (8) \quad \bar{N}_{0,i}(t) &= \bar{N}_{0,i}([0, t]), \quad \bar{N}_{0,i}(-t) = -\bar{N}_{0,i}([-t, 0]), \\ \bar{D}_i(t) &= \bar{D}_i([0, t]), \quad \bar{D}_i(-t) = -\bar{D}_i([-t, 0]), \\ \bar{D}_{i,j}(t) &= \bar{D}_{i,j}([0, t]), \quad \bar{D}_{i,j}(-t) = -\bar{D}_{i,j}([-t, 0]). \end{aligned}$$

and

$$\bar{X}_i(t) = \bar{N}_{0,i}(t) + \sum_{j:j \neq i, 1 \leq j \leq d} \bar{D}_{j,i}(t) - \bar{D}_i(t).$$

Then,  $\bar{X}_i(t)$  is a two-sided stationary process. Finally put for  $t \leq 0$ ,

$$(9) \quad \bar{Y}'(-t) = -\bar{X}(t) + \sup_{s \geq t} \bar{X}(s).$$

Observe that the for any deterministic time  $T < 0$ , the process  $\{\bar{Y}'_i(T+t) - \bar{Y}'_i(T) : 0 \leq t \leq |T|\}$  satisfies the SDE (5) only replacing the renewal processes with their respective stationary versions. We just need to show that  $Y'$  has a unique stationary distribution which is the same as the distribution of  $\bar{Y}'(0)$  and thus we have that  $\bar{Y}'$  is the time-reversed, stationary version of  $Y'$ .

**Lemma 2.** *The autonomous queue  $Y'(\cdot)$  has a unique stationary distribution and therefore  $\{\bar{Y}'(-t) : t \geq 0\}$  given by (9) is the time-reversed, stationary version of  $Y'$ .*

*Proof of Lemma 2:* We proceed with a construction procedure similar to the Loynes method. For  $t \in [0, T]$  and any  $y \in \mathbb{R}^d$  define

$$\begin{aligned} dY'_i(t) &= dN_{0,i}(t) + \sum_{j:j \neq i, 1 \leq j \leq d} dD_{j,i}(t) - I(Y'_i(t_-) > 0) dD_i(t), \\ Y'_i(0) &= y. \end{aligned}$$

We then have that

$$Y'_i(t) = (y_i + X_i(t)) - \inf_{0 \leq s \leq t} \min(y_i + X_i(s), 0)$$

and therefore

$$\begin{aligned} Y'_i(T) &= X_i(T) - \inf_{0 \leq s \leq T} \min(X_i(s), -y_i) \\ &= - \inf_{0 \leq s \leq T} \{\min(X_i(s), -y_i) - X_i(T)\} \\ &= \sup_{0 \leq s \leq T} \{\max(X_i(T) - X_i(s), y_i + X_i(T))\} \\ &= \sup_{0 \leq u \leq T} \{\max(X_i(T) - X_i(T-u), y_i + X_i(T))\}. \end{aligned}$$

As  $T \rightarrow \infty$  we have that  $X_i(T) \rightarrow -\infty$  and  $X_i(T) - X_i(T-u) \Rightarrow \bar{X}_i(u)$  as  $T \rightarrow \infty$  (weakly) and therefore  $Y'_i(T) \Rightarrow \bar{Y}'_i(0)$  regardless of the initial condition.  $\square$

Given the time-reversed, stationary version of  $Y'$ , it suffices to simulate

$$\bar{X}_i^*(t) = \sup_{r \geq t} \bar{X}_i(r),$$

jointly with  $\bar{X}_i(t)$  for all  $i \in \{1, \dots, d\}$ .

**5.2. Connection between  $\bar{X}_i^*(t)$  and Associated Random Walks.** We note that  $E[\bar{X}_i(1)] < 1$  due to (4), therefore,  $\bar{X}_i(t) \rightarrow -\infty$  as  $t \rightarrow \infty$ . Note that

$$X_i^*(t) = \max(\sup\{X_i(r) : t \leq r \leq u\}, X_i^*(u)).$$

To construct a bound for  $\bar{X}_i^*(\cdot)$ , we will construct a non-increasing process  $Z_i(\cdot)$ , such that  $Z_i(u) \geq X_i^*(u)$  and  $Z_i(u) \rightarrow -\infty$  with probability one. Since  $\sup\{X_i(r) : t \leq r \leq u\}$  is clearly non-decreasing in  $u$ , our ability to simulate  $Z_i(u)$  will allow us to sample  $X_i^*(t)$  in finite time.

**5.2.1. Construction of the Upper Bound  $Z_i(\cdot)$ .** We now give the definition of  $Z_i(\cdot)$ . Following (4), we can pick  $\bar{\delta} > 0$  small enough so that

$$\lambda_i + \sum_{j=1}^d Q_{j,i} \mu_j^0 + \bar{\delta} \left( 1 + \sum_{j=1}^d Q_{j,i} \right) < \mu_i^0.$$

Next we define  $\gamma_i = \lambda_i + \bar{\delta}$ ,  $\varphi_{j,i} = Q_{j,i}(\mu_j + \bar{\delta})$  and  $\beta_i = \gamma_i + \sum_{j=1}^d \varphi_{j,i}$ , and split

$$\bar{X}_i(t) = (\bar{N}_{0,i}(t) - \gamma_i t) + \sum_{j=1}^d (\bar{D}_{j,i}(t) - \varphi_{j,i} t) + (\beta_i t - \bar{D}_i(t))$$

so that

$$\bar{X}_i^*(t) \leq \sup_{r \geq t} (\bar{N}_{0,i}(t) - \gamma_i t) + \sum_{j=1}^d \sup_{r \geq t} (\bar{D}_{j,i}(t) - \varphi_{j,i} t) + \sup_{r \geq t} (\beta_i t - \bar{D}_i(t)).$$

Finally, we define three non-increasing processes as

$$\begin{aligned} \bar{N}_{0,i}^*(t) &= \sup_{r \geq t} (\bar{N}_{0,i}(r) - \gamma_i r), \\ \bar{D}_{j,i}^*(t) &= \sup_{r \geq t} (\bar{D}_{j,i}(r) - \varphi_{j,i} r), \\ \bar{D}_i^*(t) &= \sup_{r \geq t} (\beta_i r - \bar{D}_i(r)), \end{aligned}$$

for all  $t \geq 0$ . Observe that by the selection of  $\beta_i$ ,  $\varphi_{j,i}$ , and  $\gamma_i$ , all the three processes just defined are non-increasing and go to minus infinity with probability 1. As a result,

$$Z_i(t) := \bar{N}_{0,i}^*(t) + \sum_{j=1}^d \bar{D}_{j,i}^*(t) + \bar{D}_i^*(t) \text{ is non-increasing and goes to } -\infty \text{ as } t \rightarrow \infty.$$

Now we explain how to simulate jointly

$$(\bar{N}_{0,i}^*(t), \bar{N}_{0,i}(t), \bar{D}_{j,i}^*(t), \bar{D}_{j,i}(t), \bar{D}_i^*(t), \bar{D}_i(t) : i, j \in \{1, \dots, d\}).$$

**5.2.2. Transforming the Simulation of  $(Z(t) : t \geq 0)$  into that of the Maximum of a Multidimensional Random Walk.** Note that  $\bar{N}_{0,i}(\cdot)$  is piecewise linear with jumps, therefore it reaches its maximum only at (or right before) the times  $\{\bar{A}_i(n)\}$  when it jumps. So are  $\bar{D}_i(\cdot)$  and  $\bar{D}_{ij}(\cdot)$ . These results are formalized by the following lemma:

**Lemma 3.** For  $t \geq 0$  and assuming that  $Q_{i,j} > 0$  in the case of  $\bar{D}_{j,i}^*(t)$ , we have that

$$\begin{aligned} \bar{N}_{0,i}^*(t) &= \max \left( \bar{N}_{0,i}(t) - \gamma_i t, \sup_{n > \bar{N}_{0,i}(t)} (n - \gamma_i \bar{A}_i(n)) + 1 \right), \\ \bar{D}_i^*(t) &= \sup_{n > \bar{D}_i(t)} (\beta_i \bar{B}_i(n) - n), \\ \bar{D}_{j,i}^*(t) &= \max \left( \bar{D}_{j,i}(t) - \varphi_{j,i} t, \sup_{n > \bar{D}_{j,i}(t)} \left( \sum_{k=1}^n I(r'_j(k) = i) - \varphi_{j,i} \bar{B}_j(n) \right) + 1 \right). \end{aligned}$$

*Proof of Lemma 3:* By definition, for any  $r \geq 0$  such that  $\bar{A}_i(k) \leq r < \bar{A}_i(k+1)$ ,  $\bar{N}_{0,i}(r) = k+1$ . As a result,

$$\max_{\bar{A}_i(k) \leq r < \bar{A}_i(k+1)} (\bar{N}_{0,i}(r) - \gamma_i r) = k+1 - \bar{A}_i(k),$$

and the maximum is reached at  $r = \bar{A}_i(k)$ . As  $\bar{A}_i(\bar{N}_{0,i}(t)) \leq t < \bar{A}_i(\bar{N}_{0,i}(t) + 1)$ ,

$$\bar{N}_{0,i}^*(t) = \sup_{r \geq t} (\bar{N}_{0,i}(r) - \gamma_i r) = \sup_{t \leq r < \bar{A}_i(\bar{N}_{0,i}(t)+1)} (\bar{N}_{0,i}(r) - \gamma_i r) \vee \sup_{n > \bar{N}_{0,i}(t)} (n + 1 - \gamma_i \bar{A}_i(n)).$$

As

$$\sup_{t \leq r < \bar{A}_i(\bar{N}_{0,i}(t)+1)} (\bar{N}_{0,i}(r) - \gamma_i r) = \bar{N}_{0,i}(t) - \gamma_i t,$$

and

$$\sup_{n > \bar{N}_{0,i}(t)} (n + 1 - \gamma_i \bar{A}_i(n)) = \sup_{n > \bar{N}_{0,i}(t)} (n - \gamma_i \bar{A}_i(n)) + 1,$$

we have reach the expression for  $\bar{N}_{0,i}^*(t)$ . The same argument applies to  $\bar{D}_{j,i}^*(\cdot)$ . As to  $\bar{D}_i^*(\cdot)$ , note that

$$\sup_{\bar{B}_i(k) \leq r < \bar{B}_i(k+1)} (\beta_i r - \bar{D}_i(r)) = \beta_i \bar{B}_i(k+1) - (k+1) = \lim_{r \rightarrow \bar{B}_i(k+1)-} \beta_i r - \bar{D}_i(r),$$

and  $\bar{B}_i(\bar{D}_i(t)) \leq t < \bar{B}_i(\bar{D}_i(t) + 1)$ , therefore  $\bar{D}_i^*(t) = \sup_{n > \bar{D}_i(t)} (\beta_i \bar{B}_i(n) - n)$ .  $\square$

Therefore, to simulate the processes  $\bar{N}_{0,i}^*(\cdot)$ ,  $\bar{D}_i^*(\cdot)$  and  $\bar{D}_{ij}^*(\cdot)$ , we only need to observe the processes  $\bar{N}_{0,i}(\cdot)$ ,  $\bar{D}_i(\cdot)$  and  $\bar{D}_{ij}(\cdot)$  at the discrete times when they jump, which can be expressed as random walks. The random walks have increments  $(\bar{U}_{0,i}(n), \bar{V}_i^0, \bar{V}_{ji}^0(0))$  defined as

$$\begin{aligned} \bar{U}_{0,i}(n) &= 1 - \gamma_i(\bar{A}_i(n) - \bar{A}_i(n-1)), \quad \bar{V}_i^0(n) = \beta_i(\bar{B}_i(n) - \bar{B}_i(n-1)) - 1, \\ \bar{V}_{ji}^0(n) &= I(r'_j(n) = i) - \varphi_{j,i}(\bar{B}_j(n) - \bar{B}_j(n-1)), \text{ for } n \geq 1 \end{aligned}$$

and for  $n = 0$ ,

$$\bar{U}_i(0) = -\gamma_i \bar{A}_i(0), \quad \bar{V}_i^0(0) = \beta_i \bar{B}_i(0), \quad \bar{V}_{j,i}^0(0) = -\varphi_{j,i} \bar{B}_i(0).$$

For the pair of  $(i, j)$  with  $\varphi_{j,i} = 0$ , we have that  $\bar{V}_{j,i}^0(n) \equiv 0$  and we can ignore these coordinates. But in order to keep the notation succinct, let us denote by

$$\bar{W}_i(n) = (\bar{U}_i(n), \bar{V}_i^0(n), \bar{V}_{1,i}^0(n), \dots, \bar{V}_{d,i}^0(n))^T$$

for  $n \geq 0$ , and let

$$W(n) = (\bar{W}_1(n), \bar{W}_2(n), \dots, \bar{W}_d(n))^T.$$

Observe that  $W(n)$  is a vector of dimension  $d \times (d+2)$ . To make the notation homogeneous we write  $W_j(n)$  for the  $j$ -th coordinate of  $W(n)$  where  $1 \leq j \leq d \times (d+2)$ . Now we can define a  $d \times (d+2)$ -dimensional random walk  $S(k) = S(k-1) + W(k)$ , for  $k \geq 1$ , with  $S(0) = W(0)$ . Define its maximum process as

$$M_j(n) = \sup_{k \geq n} S_j(k) \text{ for } 1 \leq j \leq d \times (d+2).$$



Following Lemma 3, to simulate

$$(\bar{N}_{0,i}^*(t), \bar{N}_{0,i}(t), \bar{D}_{j,i}^*(t), \bar{D}_{j,i}(t), \bar{D}_i^*(t), \bar{D}_i(t) : i, j \in \{1, \dots, d\})$$

is equivalent to simulate  $(M(n), S(n) : n \geq 0)$  jointly. Fortunately, there is an algorithm that allows us to carry out this simulation problem for  $(M(n), S(n))$ , adapted from work of [1] and [3], we provide details here for completeness.

**Remark:** In the following sections we shall simulate  $(M(n) - W(0), S(n) - W(0))$ , which is equivalent to simulating the sequence  $(M(n), S(n) : n \geq 0)$  assuming that  $S(0) = 0$ . In the end, the random variable  $W(0)$  can be simulated independently from everything else.

**5.3. Sampling the Infinite Horizon Maximum of a Multidimensional Random Walk with Negative Drift.** Let  $S_{l'}(n)$  be the coordinate of the random walk corresponding to  $\bar{V}_{j,i}^0(n)$ . We have that either  $S_{l'}(n) \equiv 0$  when  $Q_{i,j} = 0$ , or  $E[S_{l'}(n)] < 0$ . For those coordinates for which  $S_{l'}(n) \equiv 0$  we have that  $M_{l'}(n) = 0$  and there is nothing to do. So, let us assume for simplicity and without loss of generality that  $E[S_i(n)] < 0$  for all  $1 \leq i \leq l = d(d+2)$ .

Define for each  $\theta \in \mathbb{R}$

$$\psi_i(\theta) = \log E[\exp(\theta_i W_i(k))],$$

and set

$$(10) \quad \begin{aligned} P_{\theta_i}(W_1(k) \in dy_1, \dots, W_l(k) \in dy_l) \\ = \frac{\exp(\theta_i y_i - \psi_i(\theta))}{E[\exp(\theta_i W_i(k))]} P(W_1(k) \in dy_1, \dots, W_d(k) \in dy_d), \end{aligned}$$

where  $\theta_i \in \mathbb{R}$  and  $E \exp(\theta_i W_i(k)) < \infty$ . Moreover, we impose the following assumption for simplicity.

**Assumption 2b):** For each  $i$  there exists  $\theta_i^*$  such that

$$\psi_i(\theta_i^*) = 0.$$

**Remark:** Assumption 2b) is a strengthening of Assumption 2. We can carry out our ideas under Assumption 2 following [1] as we explain next. First, instead of  $(M(n) : n \geq 0)$ , given a vector  $a' = (a'_1, a'_2, \dots, a'_d)^T$  with non-negative components that we will explain how to choose momentarily, consider the process  $S_{a'}(\cdot)$  and  $M_{a'}(\cdot)$  defined by

$$S_{a'}(n) := S(n) + a'n, \quad M_{a'}(n) = \max_{k \geq n} (S_a(k)).$$

Note that we can simulate  $(S(n), M(n) : n \geq 0)$  if we are able to simulate  $(S_{a'}(n), M_{a'}(n) : n \geq 0)$ . Now, note that  $\psi_i(\cdot)$  is strictly convex and that  $d\psi_i(0)/d\theta < 0$  so there exists  $a'_i > 0$  large

enough to force the existence of  $\theta_i^* > 0$  such that  $E[\exp(\theta_i^* W_i(1) + a_i' \theta_i^*)] = 1$ , but at the same time small enough to keep  $E[(W_i(1) + a_i')] < 0$ ; again, this follows by strict convexity of  $\psi_i(\cdot)$  at the origin. So, if Assumption A3b) does not hold, but Assumption A3) holds, one can then execute Algorithm 2 based on the process  $S_{a'}(\cdot)$ .

5.3.1. *Construction of  $(S(n), M(n) : n \geq 0)$  via “milestone events”.* We will describe the construction of a pair of sequences of stopping times (with respect to the filtration generated by  $(S(n) : n \geq 0)$ ), denoted by  $(\Lambda_n : n \geq 0)$  and  $(\Gamma_n : n \geq 1)$ , which track certain downward and upward milestones in the evolution of  $(S(n) : n \geq 0)$ . We follow similar steps as described in [2] and [3]. These “milestone events” will be used in the design of our proposed algorithm. The elements of the two stopping times sequences interlace with each other (when finite) and their precise description follows next.

We start by fixing any  $m > 0$ . Eventually, we shall choose  $m$  suitably large as we shall discuss in equation (18), but our conceptual discussion here is applicable to any  $m > 0$ . Now set  $\Lambda_0 = 0$ . We observe the evolution of the process  $(S(n) : n \geq 0)$  and detect the time  $\Lambda_1$  (the first downward milestone),

$$\Lambda_1 = \inf \{n \geq \Lambda_0 : S(n) < -m\mathbf{e}\},$$

where the inequality is componentwise. That is,  $S_i(n) < -m$  for all  $1 \leq i \leq l$ .

Once  $\Lambda_1$  is detected we check whether or not  $\{S(n) : n \geq \Lambda_1\}$  ever goes above the height  $S(\Lambda_1) + m$  (the first upward milestone); namely we define

$$\Gamma_1 = \inf \{n \geq \Lambda_1 : S_i(n) > m + S_i(\Lambda_1) \text{ for some } 1 \leq i \leq l\}.$$

For now let us assume that we can check if  $\Gamma_1 = \infty$  or  $\Gamma_1 < \infty$  (how exactly to do so will be explained in Section 5.3.2). To continue simulating the rest of the path, namely  $\{S(n) : n > \Lambda_1\}$ , we potentially need to keep track of the conditional upper bound implied by the fact that  $\Gamma_1 = \infty$ . To this end, we introduce the conditional upper bound variable  $C_{UB}$  (initially  $C_{UB} = \infty$ ). If at time  $\Lambda_1$  we detect that  $\Gamma_1 = \infty$ , then we set  $C_{UB} = S(\Lambda_1) + m$  and continue sampling the path of the random walk conditional on never crossing the upper bound  $S(\Lambda_1) + m$  in any of the coordinates. That is, conditional on  $\{S(n) < C_{UB} : n > \Lambda_1\}$ . Otherwise, if  $\Gamma_1 < \infty$ , we simulate the path conditional on  $\Gamma_1 < \infty$ , until we detect the time  $\Gamma_1$ . We continue on, sequentially checking whenever a downward or an upward milestone is crossed as follows: for  $j \geq 2$ , define

$$(11) \quad \begin{aligned} \Lambda_j &= \inf \{n \geq \Gamma_{j-1} I(\Gamma_{j-1} < \infty) \vee \Lambda_{j-1} : S(n) < S(\Lambda_{j-1}) - m\mathbf{e}\} \\ \Gamma_j &= \inf \{n \geq \Lambda_j : S_i(n) - S_i(\Lambda_j) > m \text{ for some } 1 \leq i \leq l\}, \end{aligned}$$

with the convention that if  $\Gamma_{j-1} = \infty$ , then  $\Gamma_{j-1}I(\Gamma_{j-1} < \infty) = 0$ . Therefore, we have that  $\Gamma_{j-1}I(\Gamma_{j-1} < \infty) > \Lambda_{j-1}$  if and only if  $\Gamma_{j-1} < \infty$ .

Let us define

$$(12) \quad \Delta = \inf\{\Lambda_n : \Gamma_n = \infty, n \geq 1\}.$$

So, for example, if  $\Gamma_1 = \infty$  we have that  $\Delta = \Lambda_1$  and the drifted random walk will never reach level  $S(\Lambda_1) + m < S(0)$  again. This allows us to evaluate  $M(0)$  by computing

$$(13) \quad M(0) = \max\{S(n) : 0 \leq n \leq \Delta\},$$

the maximum is taken over  $n$  for each coordinate.

Similarly, the event  $\Gamma_j = \infty$ , for some  $j \geq 1$ , implies that the level  $S_i(\Lambda_j) + m$  is never crossed for any  $i$  (that is  $S_i(n) \leq S_i(\Lambda_j) + m$ ) for all  $n \geq \Lambda_j$ , and we let  $C_{UB} = S(\Lambda_j) + m$ . The value of the vector  $C_{UB}$  keeps updating as the random walk evolves, at times where  $\Gamma_j = \infty$ .

The advantage of considering these stopping times is the following: once we observed that some  $\Gamma_j = \infty$ , the values of  $\{M_i(n) : n \leq \Gamma_{j-1}1(\Gamma_{j-1} < \infty) \vee \Lambda_{j-1}\}$  for each  $1 \leq i \leq l$  are known without a need of further simulation. Proposition 1 ensures that it suffices to sequentially simulate  $(\Lambda_n : n \geq 0)$  and  $(\Gamma_n : n \geq 1)$  jointly with the underlying random walk in order to sample from the sequence  $(S(n), M(n) : n \geq 0)$ . The proof of Proposition 1 is easily adapted from the one dimensional case discussed in [3] and thus it is omitted.

**Proposition 1.** *Set  $\Lambda_0 = 0$  and let  $(\Lambda_n : n \geq 1)$  and  $(\Gamma_n : n \geq 1)$  be as (11). We have that*

$$(14) \quad P_0(\lim_{n \rightarrow \infty} \Lambda_n = \infty) = 1 \quad \text{and} \quad P_0(\Lambda_n < \infty) = 1, \quad \forall n \geq 1.$$

Furthermore,

$$(15) \quad P_0(\Gamma_n = \infty, \text{ i.o. }) = 1.$$

In the setting of Proposition 1, for each  $k \geq 0$  we can define  $N_0(k) = \inf\{n \geq 1 : \Lambda_n \geq k\}$  and  $\mathcal{T}(k) = \inf\{j \geq N_0(k) + 1 : \Gamma_j = \infty\}$ . Both of them are finite random variables such that

$$(16) \quad M(k) = \max\{S(n) : k \leq n \leq \Lambda_{\mathcal{T}(k)}\}$$

In other words,  $\Lambda_{\mathcal{T}(k)}$  is the time, not earlier than  $k$ , at which we detect a second unsuccessful attempt at building an upward patch directly. The fact that the relation in (16) holds, follows easily by construction of the stopping times in (11). Note that it is important, however, to define  $\mathcal{T}(k) \geq N_0(k) + 1$  so that  $\Lambda_{N_0(k)+1}$  is computed first. In that way,

we can make sure that the maximum of the sequence  $(S(n) : n \geq k)$  is achieved between  $k$  and  $\Lambda_{\mathcal{T}(k)}$ .

These observation gives rise to our suggested high-level scheme. The procedure sequentially constructs the random walk in the intervals  $[\Lambda_{n-1}, \Lambda_n)$  for  $n \geq 1$ . Here is the high-level procedure to construct  $(S(n), M(n) : n \geq 0)$ :

**Algorithm 1.** *At the  $k$ -th iteration, for  $k \geq 1$ :*

**Step 1: “downward patch”.** *Conditional on the path not crossing  $C_{UB}$  we simulate the path until we detect  $\Lambda_k$ , which is the first time when the random walk visits the set  $(-\infty, S_1(\Lambda_{k-1}) - m] \times \dots \times (-\infty, S_l(\Lambda_{k-1}) - m]$ .*

**Step 2: “upward patch”.** *Check whether or not the level  $S_i(\Lambda_k) + m$  is ever crossed by any of the coordinates  $i$ . That is, whether  $\Gamma_k < \infty$  or not. If the answer is “Yes” then, conditional on the path crossing  $S_i(\Lambda_k) + m$  for some  $i$ , but not crossing the level  $(C_{UB})_i$ , we simulate the path until we detect  $\Gamma_k$ , the first time the level  $S_i(\Lambda_k) + m$  for at least one of the coordinates  $i$ . Otherwise ( $\Gamma_j = \infty$ ), and we can update  $C_{UB}$ :  $C_{UB} \leftarrow S(\Lambda_j) + m\mathbf{e}$*

The implementation of the steps in Algorithm 1 will be discussed in detail in the next sections, culminating with the precise description given in Algorithm 4 at the end of Section 5.3.3.

**5.3.2. Sampling  $M(0)$  jointly with  $(S(1), \dots, S(\Delta))$ .** The goal of this section is to sample exactly from  $M(0)$ . To this end we need to simulate the sample path up to the first  $\Gamma_j$  such that  $\Gamma_j = \infty$  (recall that  $\Delta$  was defined to be the corresponding  $\Lambda_j$ ). This sample path will be used in the construction of further steps in Algorithm 1. This construction is directly taken from [1].

For any positive vectors  $a, b > 0$ . Let

$$(17) \quad \begin{aligned} \tau_b &= \inf \{n \geq 0 : S_i(n) > b_i \text{ for some } i\}, \\ \tau_{-b} &= \inf \{n \geq 0 : S_i(n) < -b_i \text{ for all } i\}, \\ P_a(\cdot) &= P(\cdot \mid S(0) = a). \end{aligned}$$

Since we concentrate on  $M(0)$ , we have that  $C_{UB} = \infty$ . We first need to explain a procedure to generate a Bernoulli random variable with success parameter  $P_0(\tau_{me} < \infty)$ , for suitably chosen  $m > 0$ . Also, this procedure, as we shall see, will allow us to simultaneously simulate  $(S(1), \dots, S(\tau_{me}))$  given that  $\tau_{me} < \infty$ .

We think of the probability measure  $P_0(\cdot)$  as defined on the canonical space  $\Omega = \{0\} \times \mathbb{R}^l \times \mathbb{R}^l \times \dots$  endowed with  $\sigma$ -field generated by the Borel  $\sigma$ -field of finite dimensional projections (i.e. the Kolmogorov  $\sigma$ -field). Our goal is to simulate from the conditional law

of  $(S(n) : 0 \leq n \leq \tau_{me})$  given that  $\tau_{me} < \infty$  and  $S(0) = 0$ , which we shall denote by  $P_0^*$  in the rest of this part.

First, we select  $m > 0$  such that

$$(18) \quad \sum_{k=1}^l \exp(-\theta_i^* m) < 1.$$

Now let us introduce our proposal distribution  $P'_0(\cdot)$ , defined on the space  $\Omega' = \Omega \times \{1, 2, \dots, l\}$ . We endow the probability space with the associated Kolmogorov  $\sigma$ -field. So, a typical element  $\omega'$  sampled under  $P'_0(\cdot)$  is of the form  $\omega' = ((S(n) : n \geq 0), \text{Index})$ , where  $\text{Index} \in \{1, 2, \dots, l\}$ . The distribution of  $\omega'$  induced by  $P'_0(\cdot)$  is described as follows, first,

$$(19) \quad P'_0(\text{Index} = i) = w_i := \frac{\exp(-\theta_i^* m)}{\sum_{j=1}^l \exp(-\theta_j^* m)}.$$

Now, given  $\text{Index} = i$ , for every set  $A \in \sigma(S(k) : 0 \leq k \leq n)$ ,

$$P'_0(A | \text{Index} = i) = E_0[\exp(\theta_i^* Z_i(t)) I_A].$$

In particular, the Radon-Nikodym derivative (i.e. the likelihood ratio) between the distribution of  $\omega = (S(k) : 0 \leq k \leq n)$  under  $P'_0(\cdot)$  and  $P_0(\cdot)$  is given by

$$\frac{dP'_0}{dP_0}(\omega) = \sum_{i=1}^l w_i \exp(\theta_i^* S_i(n)).$$

The distribution of  $(S(k) : k \geq 0)$  under  $P'_0(\cdot)$  is precisely the proposal distribution that we shall use to apply acceptance / rejection. It is straightforward to simulate under  $P'_0(\cdot)$ . First, sample  $\text{Index}$  according to the distribution (19). Then, conditional on  $\text{Index} = i$ , the process  $S(\cdot)$  is also a multidimensional random walk. Indeed, given  $\text{Index} = i$ , under  $P'_0(\cdot)$  it follows that  $S(n)$  can be represented as

$$(20) \quad S(n) = W'(1) + \dots + W'(n),$$

where  $W'(k)$ 's are i.i.d. with distribution obtained by exponential tilting, such that for all  $A \in \sigma(W'(k))$ ,

$$(21) \quad P'_0(W'(k) \in A) = E[\exp(\theta_i^* W_i) I_A].$$

Now, note that we can write

$$\begin{aligned} E'_0(S_{Index}(n)) &= \sum_{i=1}^l E_0(S_i(n) \exp(\theta_i^* S_i(n))) P'_0(Index = i) \\ &= \sum_{i=1}^l \frac{d\psi_i(\theta_i^*)}{d\theta} w_i > 0, \end{aligned}$$

where the last inequality follows by convexity of  $\psi_k(\cdot)$  and by definition of  $\theta_k^*$ . So, we have that  $S_{Index}(n) \nearrow \infty$  as  $n \nearrow \infty$  with probability one under  $P'_0(\cdot)$ , by the Law of Large Numbers. Consequently  $\tau_{me} < \infty$  a.s. under  $P'_0(\cdot)$ .

Recall that  $P_0^*(\cdot)$  is the conditional law of  $(S(n) : 0 \leq n \leq \tau_{me})$  given that  $\tau_{me} < \infty$  and  $S(0) = 0$ . In order to assure that we can indeed apply acceptance / rejection theory to simulate from  $P_0^*(\cdot)$ , we need to show that the likelihood ratio  $dP_0/dP'_0$  is bounded. Indeed,

$$\begin{aligned} \frac{dP_0^*}{dP'_0}(S(n) : 0 \leq t \leq \tau_{me}) &= \frac{1}{P_0(\tau_{me} < \infty)} \times \frac{dP_0}{dP'_0}(S(n) : 0 \leq t \leq \tau_{me}) \\ (22) \quad &= \frac{1}{P_0(\tau_{me} < \infty)} \times \frac{1}{\sum_{i=1}^l w_i \exp(\theta_i^* S_i(\tau_{me}))}. \end{aligned}$$

Upon  $\tau_{me}$ , there is an index  $I'$  ( $I'$  may be different from  $Index$ ) such that  $\exp(\theta_{I'}^* S_{I'}(\tau_{me})) \geq \exp(\theta_{I'}^* m)$ , therefore

$$(23) \quad \frac{1}{\sum_{i=1}^l w_i \exp(\theta_i^* S_i(\tau_{me}))} \leq \frac{1}{w_{I'} \exp(\theta_{I'}^* m)} = \sum_{i=1}^l \exp(-\theta_i^* m) < 1,$$

where the last inequality follows by (18). Consequently, plugging (23) into (22) we obtain that

$$(24) \quad \frac{dP_0^*}{dP'_0}(S(n) : 0 \leq n \leq \tau_{me}) \leq \frac{1}{P_0(\tau_{me} < \infty)}.$$

Now we are ready to fully discuss our algorithm to sample  $J$  and  $\omega = (S(1), \dots, S(\tau_{me}))$  given  $\tau_{me} < \infty$ . Upon termination we will output the pair  $(J, \omega)$ . If  $J = 1$ , then we set  $\omega = (S(1), \dots, S(\tau_{me}))$ . Otherwise ( $J = 0$ ), we set  $\omega = []$ , the empty vector.

**Algorithm 2.** INPUT:  $\theta_i^*$  and  $m$  satisfying (18).

OUTPUT:  $J \sim \text{Ber}(P_0(\tau_{me} < \infty))$  and  $\omega$ . If  $J = 1$ , then  $\omega = (S(1), \dots, S(\tau_{me}))$ . Otherwise ( $J = 0$ ),  $\omega = []$

Step 1: Sample  $(S(n) : 0 \leq t \leq \tau_{me})$  according to  $P'_0(\cdot)$  as indicated via equations (20) and (21).

Step 2: Given  $(S(n) : 0 \leq t \leq \tau_{me})$ , simulate a Bernoulli  $J$  with probability

$$\frac{1}{\sum_{i=1}^l w_i \exp(\theta_i^* S_i(\tau_{me}))}.$$

Step 3: If  $J = 1$ , **output**  $(J, \omega)$ , where  $\omega = (S(j) : 1 \leq j \leq \tau_{me})$ . ELSE, if  $J = 0$ , **output**  $(J, \omega)$ , where  $\omega = []$ .

The authors in [1] show that the output of the previous procedure indeed follows the distribution of  $(S(n) : 0 \leq n \leq \tau_{me})$  given that  $\tau_{me} < \infty$  and  $S(0) = 0$ . Moreover, the Bernoulli random variable  $J$  has probability  $P_0(\tau_{me} < \infty)$  of success.

Now we are ready to give the algorithm sampling  $M(0)$  jointly with  $(S(1), \dots, S(\Delta))$ . Before we move on to the algorithm let us define the following. Given a vector  $\mathbf{s}$ , of dimension  $d \geq 1$ , we let  $\mathbf{L}(\mathbf{s}) = \mathbf{s}(d)$  (i.e. the  $d$ -th component of the vector  $\mathbf{s}$ ).

**Algorithm 3.** INPUT Same as Algorithm 2

OUTPUT The path  $(S(1), \dots, S(\Delta))$

Initialization  $\mathbf{s} \leftarrow []$ ,  $F \leftarrow 0$ , and  $\mathbf{L} = 0$ .

(Initially  $\mathbf{s}$  is the empty array, the variable  $\mathbf{L}$  represents the last position of the drifted random walk.

WHILE  $F = 0$

Sample  $(S(1), \dots, S(\tau_{-2me}))$  given  $S(0) = 0$ ,

$\mathbf{s} = [\mathbf{s}, \mathbf{L} + S(1), \dots, \mathbf{L} + S(\tau_{-2me})]$ ,

$\mathbf{L} = \mathbf{L} + S(\tau_{-2me})$ .

Call Algorithm 2 and obtain  $(J, \omega)$ ,

IF  $J = 1$  Set  $\mathbf{s} = [\mathbf{s}, \mathbf{L} + \omega]$ ,

ELSE  $F \leftarrow 1$  ( $J = 0$ )

END WHILE

OUTPUT  $\mathbf{s}$ .

**Proposition 2.** The output of Algorithm 3 has the correct distribution according to (12) and (13). Moreover, if  $\bar{N}$  is the number of random variables needed to terminate Algorithm 3, there is  $\delta > 0$  such that  $E[\exp(\delta \bar{N})] < \infty$ .

*Proof of Proposition 2:* As noted earlier, this follows directly from the analysis in [1].  $\square$

5.3.3. From  $M(0)$  to  $(S(k), M(k) : 0 \leq k \leq n)$ . In this section we will explain in detail the complete procedure to sample  $M(k)$  jointly with  $S(k)$  for  $1 \leq k \leq n$ , where  $n$  is a

finite number given by the user. The algorithm is similar as that for sampling  $(M(0)$  and  $S(1), \dots, S(\Delta)$  and is also based on simulating the downward and upward patches. The main difference is that  $C_{UB} < \infty$  for  $M(k)$  with  $k > 0$  and hence we need to simulate the random walk  $S(k)$  conditional on that it never crosses the level  $C_{UB}$ . In particular, we shall use the algorithm for sampling  $M(0)$  developed in Section 5.3.2 to help us simulate the conditional probability.

In Step 1 we need to sample the maximum of the drifted random walk  $(S(n) : n \geq 0)$ . Suppose that our current position is  $S(\Lambda_j)$  and we know that the random walk will never reach position  $C_{UB}$ . In other words, there exist some  $n \leq j - 1$  such that  $\Gamma_n = \infty$ . Let  $i = \max\{1 \leq n \leq j - 1 : \Gamma_n = \infty\}$ , then  $C_{UB} = S(\Lambda_i) + m$ . We now explain how to simulate the path up to the first time  $\Lambda_{\bar{n}}$ , for  $\bar{n} > j$ , such that  $\Gamma_{\bar{n}} = \infty$ .

First, we call Algorithm 3 and obtain the output  $\omega = (s_1, \dots, s_\Delta)$ . We compute  $M(0)$  according to (13) and keep calling Algorithm 3 until we obtain  $M(0) \leq C_{UB} - S(\Lambda_j)$ , at which point we set

$$(25) \quad (S(\Lambda_j), S(\Lambda_j + 1), \dots, S(\Lambda_{\bar{n}})) = (S(\Lambda_j), S(\Lambda_j) + s_1, \dots, S(\Lambda_j) + s_\Delta).$$

It is clear from the construction of the path that indeed  $\omega = (s_1, \dots, s_\Delta)$  has the correct distribution of  $(S(1), \dots, S(\Delta))$  given  $\tau_{C_{UB}-S(\Lambda_j)} = \infty$  and  $S(0) = 0$ . Then, we simply update  $C_{UB} \leftarrow S(\Lambda_j) + s_\Delta + m\mathbf{e}$ .

We close this section by giving the explicit implementation of our general method outlined in Subsections 5.3.1. In order to describe the procedure, let us recall some definitions. Given an array  $\mathbf{s}$  of dimensions  $l \times n \geq 1$ , let  $\mathbf{L}(\mathbf{s}) = \mathbf{s}(n)$  (the last column vector of dimension  $l$  in the array). Given an array  $\mathbf{z}$  of size  $l' \times n$ , set  $\mathbf{d}(\mathbf{z}) = n$  (the number of columns in the array). We shall evaluate  $\mathbf{d}(\cdot)$  on arrays that might have different numbers of rows.

**Algorithm 4.** *INPUT Same as Algorithm 2*

*OUTPUT  $(S(k), M(k) : 0 \leq k \leq n)$*

*Initialization  $\mathbf{s} \leftarrow [0]$ ,  $C_{UB} \leftarrow \infty$ ,  $\mathbf{N} \leftarrow []$ ,  $F \leftarrow 0$ . (Initialize the sample path with the array containing only one vector of  $l$ -dimensions.)*

*Comments: The vector  $\mathbf{N}$ , which is initially empty records the times  $\Lambda_j$  such that  $\Gamma_j = \infty$ .  $F$  is a Boolean variable which detects when we have enough information to compute  $M(n)$*

*WHILE  $F = 0$*

*$F_1 \leftarrow 0$*

*WHILE  $F_1 = 0$*

*Call Algorithm 3. Obtain as output  $\omega = (s_1, \dots, s_\Delta)$ , and get  $M(0)$ .*

*IF  $M(0) \leq C_{UB} - \mathbf{L}(\mathbf{s})$ , update  $C_{UB} = \mathbf{L}(\mathbf{s}) + s_\Delta + M(0)\mathbf{e}$ ,  $\mathbf{s} = [\mathbf{s}, \mathbf{L}(\mathbf{s}) + \omega]$ ,  $\mathbf{N} = [\mathbf{N}, \mathbf{d}(\mathbf{s})]$  and  $F_1 = 1$ .*



END WHILE

IF  $\mathbf{N}(\mathbf{d}(\mathbf{N}) - 1) \geq n$ , set  $F \leftarrow 1$ .

END WHILE

FOR  $k = 0, \dots, n$

$M(k) = \max(\mathbf{s}(k+1), \mathbf{s}(k+2), \dots, \mathbf{s}(\mathbf{d}(\mathbf{s})))$ ,

$S(k) = \mathbf{s}(k+1)$ .

END FOR

**OUTPUT:**  $(S(k), M(k) : 1 \leq k \leq n)$ .

## 6. NUMERICAL RESULTS

To test the numerical performance and correctness of our algorithm, we implement our algorithm in Matlab. In particular, we consider a 2-station Jackson network with Poisson arrivals and exponential service times, so that the true value of the steady-state distribution is known in closed form. In the numerical test, we shall fix the routing matrix  $Q = [0, 0.11; 0.1, 0]$  and run the simulation algorithm for different arrival and service rates  $\lambda$  and  $\mu$ . For each pair of  $(\lambda, \mu)$ , we generate 10000 i.i.d. samples of the number of customers  $(Y_1(\infty), Y_2(\infty))$ .

We estimate the steady-state expectation  $E[Y_i(\infty)]$  and the correlation coefficient of  $Y_1(\infty)$  and  $Y_2(\infty)$  based on the 10000 i.i.d. samples. Since the 2-station system is a Jackson network, the theoretic steady-state distribution of  $Y_i(\infty)$  is known and the true value of  $E[Y_i(\infty)] = \phi/(\mu - \phi)$ . Moreover, the true value of the correlation coefficient is 0 as the joint distribution of  $(Y_1(\infty), Y_2(\infty))$  is of product form. In Table 1, for different  $\mu$  and  $\lambda$ , we report the simulation estimations and compare them with the true values. In detail, we report the 95% confidence interval of  $E[Y_i(\infty)]$  estimated from the simulated samples. For the correlation, we report the sample correlation coefficient and the  $p$ -value of the hypothesis test that the two population are not correlated.

Figure 1 and 2 compares the histogram of the 10000 simulation samples with the true steady state distribution for two different values of  $\lambda$  and  $\mu$ . In both two cases, we can see that the empirical distribution of the i.i.d. simulated samples is very close to the true distribution.

TABLE 1. Simulation Estimation of  $E[Y_1(\infty)]$ ,  $E[Y_2(\infty)]$  and  $\text{Corr}(Y_1(\infty), Y_2(\infty))$  for different  $\lambda$  and  $\mu$ . ( $p$ -value  $> 5\%$  means no significant correlation)

Parameters					
$\lambda$	(0.2250, 0.7170)	(0.2200, 0.7670)	(0.2180, 0.7870)	(0.2160, 0.8070)	(0.2140, 0.8270)
$\mu$	(1.0000, 1.0000)	(1.0000, 1.0000)	(1.0000, 1.0000)	(1.0000, 1.0000)	(1.0000, 1.0000)
$E[Y_1(\infty)]$					
TrueValue	0.4286	0.4286	0.4286	0.4286	0.4286
Simulation	0.4265 $\pm$ 0.0152	0.4204 $\pm$ 0.0150	0.4247 $\pm$ 0.0150	0.4376 $\pm$ 0.0153	0.4228 $\pm$ 0.0155
$E[Y_2(\infty)]$					
TrueValue	3.0000	4.0000	4.5556	5.2500	6.1429
Simulation	2.9355 $\pm$ 0.0676	4.0468 $\pm$ 0.0877	4.5844 $\pm$ 0.0984	5.3057 $\pm$ 0.1156	6.1620 $\pm$ 0.1291
$\text{Corr}(Y_1(\infty), Y_2(\infty))$					
Simulation	-0.0058	-0.0128	0.0151	0.0011	0.0116
$p$ -value	55.96%	19.90%	13.13%	91.13%	24.80%

## 7. APPENDIX: TECHNICAL PROOFS

**7.1. Technical Lemmas and the Proof of Theorem 1 .** Part i) of Theorem 1 is a restatement of Lemma 4.2 in [5]. Then, part ii) follows from the next lemma.

**Lemma 4.** *Suppose we start the coupled systems  $\mathcal{N}^+$  and  $\mathcal{N}'$  empty from time 0. Then for any  $t > 0$ , when  $Y_i'(t) = y_i$ , then the service station  $i$  in system  $\mathcal{N}^+$  must be one of the three following cases:*

- (1)  $Y_i^+(t) = y_i + 1$  and the server is in service
- (2)  $Y_i^+(t) = y \in \{1, 2, \dots, y_i\}$  and the server is either in service or vacation
- (3)  $Y_i^+(t) = 0$  and the server is in vacation.

*Proof of Lemma 4:* The result follows directly by comparing the evolution of  $Y_i'(\cdot)$  given by (5), against the evolution of the number of customers waiting in the  $i$ -th queue of  $\mathcal{N}^+$ , namely  $\hat{Y}_i^+(\cdot)$ , which satisfies (6). The equations are monotone with respect to the input process, which is strictly smaller for the network  $\mathcal{N}^+$  compared to  $\mathcal{N}'$  because

$$\sum_{j: j \neq i, 1 \leq j \leq d} I(\hat{S}_j^+(t_-) > 0) dD_{j,i}(t) \leq \sum_{j: j \neq i, 1 \leq j \leq d} dD_{j,i}(t).$$

So, we conclude that  $\hat{Y}_i^+(t) \leq Y_i'(t)$ . Therefore, if  $Y_i'(t) = y_i$ , we have  $\hat{Y}_i^+(t) \leq y_i$ . As  $\hat{Y}_i^+(t)$  is the number of customers who are waiting for entering service, we can conclude

that either  $Y_i^+(t) = \widehat{Y}_i^+(t) + 1$  if the server is in service and  $Y_i^+(t) = \widehat{Y}_i^+(t)$  if the server is on vacation, and hence we are done.  $\square$

In order to prove part iii) of Theorem 1 we introduce some notation.

Let  $y = (y_1, \dots, y_d)$  be a fixed vector in  $\mathbb{N}_0^d = \{0, 1, \dots\}^d$ . Consider three vacation networks  $\mathcal{N}^+, \mathcal{N}^{++}, \mathcal{N}^{+-}$  that have the same network topology and are driven by the same arrival and activity sequences, namely,  $(A_i(n) : n \geq 0)$  and  $(v_i^0(n), r_i(n) : n \geq 1)$ , except for their initial state at time 0. In particular, we set  $Y_i^{++}(0) = y_i + 1$  with all servers in service for all  $i$ , and  $Y_i^{+-}(0) = 0$  with all servers in vacation. The state of each service station in  $\mathcal{N}^+$  at time 0 is of any one of three cases as described in Lemma 4. More precisely, we have the following system of SDEs for  $\widehat{Y}_i^{++}, \widehat{S}_i^{++}$ , and  $Y_i^{++}$  with  $i \in \{1, \dots, d\}$ ,

$$\begin{aligned} d\widehat{Y}_i^{++}(t) &= dN_{0,i}(t) + \sum_{j:j \neq i, 1 \leq j \leq d} I(\widehat{S}_j^{++}(t_-) > 0) dD_{j,i}(t) - I(\widehat{Y}_i^{++}(t_-) > 0) dD_i(t), \\ d\widehat{S}_i^{++}(t) &= (I(\widehat{Y}_i^{++}(t_-) > 0) - I(\widehat{S}_i^{++}(t_-) > 0)) dD_i(t), \\ \widehat{Y}_i^{++}(0) &= y_i + 1, \widehat{S}_i^{++}(0) = 1, \\ Y_i^{++}(t) &= \widehat{Y}_i^{++}(t) + \widehat{S}_i^{++}(t). \end{aligned}$$

The SDEs for  $\widehat{Y}_i^+, \widehat{S}_i^+, Y_i^+$ , and  $\widehat{Y}_i^{+-}, \widehat{S}_i^{+-}, Y_i^{+-}$  are exactly the same, except for the boundary conditions. In particular,  $\widehat{Y}_i^+(0) = y_i$ ,  $\widehat{S}_i^{++}(0) \in \{0, 1\}$ , and  $\widehat{Y}_i^{+-}(0) = 0$ ,  $\widehat{S}_i^{+-}(0) = 0$ . Then we have the following comparison result which implies part iii) of Theorem 1:

**Lemma 5.** *In order to distinguish servers whenever there might be ambiguity we shall call the server of the  $i$ -th service station in  $\mathcal{N}^{++}$  the server  $i^+$ , the server of the  $i$ -th station in  $\mathcal{N}^+$  is called server  $i$ , and the  $i$ -th server in  $\mathcal{N}^{+-}$  is called  $i^-$ . We claim that the following three statements hold for all servers  $i, i^+$ , and  $i^-$  ( $i = 1, 2, \dots, d$ ) and at any  $t \geq 0$  (analogous statements to 2. and 3. hold replacing  $i^+$  by  $i$  and  $i$  by  $i^-$ )*

- (1)  $Y_i^{++}(t) \geq Y_i^+(t) \geq Y_i^{+-}(t)$ .
- (2) If  $Y_i^{++}(t) = Y_i^+(t)$ , server  $i^+$  and server  $i$  are both in service or both in vacation. Similarly,  $Y_i^+(t) = Y_i^{+-}(t)$ , server  $i$  and server  $i^-$  are both in service or both in vacation
- (3) If server  $i^+$  is in vacation, server  $i$  is also in vacation. Similarly, if  $i$  is in vacation, server  $i^-$  is also in vacation.

*Proof of Lemma 5:* Let's first prove  $Y_i^{++}(t) \geq Y_i^+(t)$ .

First let's check if the claim is true for  $t = 0$ . Note that  $Y_i^{++}(0) = y_i + 1$  and  $Y_i^+ \in \{0, 1, \dots, y_i + 1\}$  and hence Statement (1) holds. As server  $i^+$  is in service at time 0, Statement (2) also hold. Finally, if  $Y_i^+(0) = Y_i^{++}(0) = y_i + 1$ , service station  $i$  is of the first case in Lemma 4 which means both server  $i$  and  $i^+$  are at service. In summary, the claim is true

for  $t = 0$ .

Let  $E(t) = \sum_{i=1}^d N_{0,i}(t) + \sum_{i=1}^d D_i(t)$  be the counting process of events that occur in the network. Define  $t(n) = \inf\{t \geq 0 : E(t) = n\}$  to be the time at which the  $n$ -th event occurs for  $n \geq 1$  and set  $t(0) = 0$ . We shall prove the statements 1-3 only for  $i$  and  $i^+$  first by induction on  $n \geq 0$  at times  $t(n)$ , since there are no changes inside the network population between two event epochs. We have verified that statements 1-3 are valid at  $t(0)$ . Assume by induction hypothesis, that statements 1-3 hold for  $t(n-1)$ , we need to consider several cases at time  $t(n)$ .

*Case 1:*  $t(n)$  corresponds to an arrival from  $N_{0,i}(\cdot)$ .

In this case,  $Y_i^{++}(t(n)) = Y_i^{++}(t(n-1)) + 1$  and  $Y_i^+(t(n)) = Y_i^+(t(n-1)) + 1$ . According to the dynamics of vacation system, a new arrival from  $N_{0,i}(\cdot)$  does not change the type of activity that is going on in servers  $i^+$  and  $i$ . So statements 1-3 hold for server  $i$  and  $i^+$  at  $t(n)$ . As to all the other servers, there are no changes between  $t(n-1)$  and  $t(n)$ . In summary, Statement 1-3 hold for all servers at  $t(n)$ .

*Case 2:*  $t(n)$  corresponds to an arrival from  $D_i(\cdot)$  and  $Y_i^{++}(t(n-1)) = Y_i^+(t(n-1))$ .

By induction hypothesis, server  $i$  and  $i^+$  are in the same type of activity at time  $t(n-1)$ . Suppose that both servers  $i$  and  $i^+$  are at vacation at  $t(n-1)$ , it is clear from the dynamics that  $Y_i^{++}(t(n)) = Y_i^+(t(n))$ . If  $Y_i^{++}(t(n)) > 0$ , it means that there was someone waiting and therefore at time  $t(n)$ , coming from vacation, now both  $i^+$  and  $i$  are now in service at time  $t(n)$ ; otherwise, from the same logic,  $Y_i^{++}(t(n)) = 0$ , implies that both  $i^+$  and  $i$  are on vacation at  $t(n)$ . Besides, there are no changes on other servers between  $t(n-1)$  and  $t(n)$ , because at  $t(n-1)$  the servers were on vacation. Therefore, Statement 1-3 hold for all servers at  $t(n)$ .

If both server  $i$  and  $i^+$  are in service at  $t(n-1)$  and  $Y_i^{++}(t(n-1)) = Y_i^+(t(n-1)) = 1$ , then  $Y_i^{++}(t(n)) = Y_i^+(t(n)) = 0$  and both server  $i^+$  and  $i$  are in vacation at  $t(n)$ . Let  $j = r_i(B_i(t(n)))$ . If  $j = 0$ , there are no changes on other servers between  $t(n-1)$  and  $t(n)$ , so Statement 1-3 hold for all servers at  $t(n)$ . If  $j > 1$ , then we can apply the argument of *Case 1* to server  $j$ ,  $j^+$ , and there are no changes on the rest servers other than  $i^+$ ,  $i$ ,  $j^+$  and  $j$ . So statements 1-3 hold for all servers at  $t = t(n)$ .

If both server  $i$  and  $i^+$  are in service at  $t(n-1)$  and  $Y_i^{++}(t(n-1)) = Y_i^+(t(n-1)) > 1$ , the argument is similar to when  $Y_i^{++}(t(n-1)) = Y_i^+(t(n-1)) = 1$  except that now both

server  $i^+$  and  $i$  are in service at  $t(n)$ .

*Case 3:*  $t(n)$  corresponds to an arrival from  $D_i(\cdot)$  and  $Y_i^{++}(t(n-1)) > Y_i^+(t(n-1))$

If server  $i^+$  is in vacation at  $t(n-1)$ , by induction hypothesis, server  $i$  is also in vacation at  $t(n-1)$ . Then, there are no changes on all other servers. Besides,  $Y_i^{++}(t(n)) = Y_i^{++}(t(n-1))$  and  $Y_i^+(t(n)) = Y_i^+(t(n-1))$  and hence  $Y_i^{++}(t(n)) > Y_i^+(t(n))$ . As  $Y_i^{++}(t(n)) > 0$ , server  $i^+$  is in service at  $t(n)$  and hence we do not contradict statement 3 for servers  $i^+$  and  $i$  at time  $t(n)$ . In summary, we conclude that Statements 1-3 hold for all servers at time  $t(n)$ .

If server  $i^+$  is in service and  $Y_i^{++}(t(n-1)) = 1$  (and so  $Y_i^+(t(n-1)) = 0$ ),  $Y_i^{++}(t(n)) = Y_i^+(t(n)) = 0$  and server  $i^+$  and  $i$  are both in vacation at time  $t(n)$ . Let  $j = r_i(D_i(t(n)))$ . If  $j = 0$ , there are no changes on all other servers and hence statements 1-3 hold for all servers at  $t(n)$ . Otherwise, we have  $Y_j^{++}(t(n)) = Y_j^{++}(t(n-1)) + 1$  and  $Y_j^+(t(n)) = Y_j^+(t(n-1))$ , as  $Y_j^{++}(t(n-1)) \geq Y_j^+(t(n-1))$  by induction hypothesis,  $Y_j^{++}(t(n)) > Y_j^+(t(n))$ . So statement 1-2 hold for server  $j^+$  and  $j$ . The type of activity that occurs in server  $j^+$  and  $j$  remains the same what was going on at time  $t(n-1)$  and hence statement 3 holds. Since there are no changes on the rest servers other than  $i^+$ ,  $i$ ,  $j^+$  and  $j$ , statement 1-3 hold at time  $t(n)$  for all servers.

If server  $i^+$  is in service at  $t(n-1)$  and  $Y_i^{++}(t(n-1)) > 1$ ,  $Y_i^{++}(t(n)) = Y_i^{++}(t(n-1)) - 1 > 0$  and server  $i^+$  is still in service at  $t(n)$ . So statement 3 holds for servers  $i^+$  and  $i$  at time  $t(n)$ . As  $Y_i^{++}(t(n-1)) \geq Y_i^+(t(n-1)) + 1$  and  $Y_i^+(t(n)) \geq Y_i^+(t(n-1))$ ,  $Y_i^{++}(t(n)) \geq Y_i^+(t(n))$  and statement 1 holds for server  $i$ . In case  $Y_i^{++}(t(n)) = Y_i^+(t(n))$ ,  $Y_i^+(t(n)) > 0$  and hence both server  $i^+$  and  $i$  are in service at  $t(n)$  and statement 2 holds. Following a similar argument as when server  $i^+$  is in service at  $t(n-1)$  and  $Y_i^+(t(n-1)) = 1$ , we can check that statement 1-3 hold for all the other servers. As a result, we can conclude that statement 1-2 hold at time  $t(n)$  for all servers.

By induction, and by the nature of the processes, which changes only at times  $t(n)$ , statements 1-3 hold for all  $t \geq 0$ .

To prove that  $Y_i^+(t) \geq Y_i^{+-}(t)$ , we can use the same induction arguments simply replacing  $Y_i^{++}(t)$  with  $Y_i^+(t)$ , and  $Y_i^+(t)$  with  $Y_i^{+-}(t)$  in statements 1-3. The induction part is exactly the same, so we are done if we can check that the three statements all hold at

time  $t(0)$ .

As  $Y_i^{+-}(0) = 0$  and all servers  $i^-$  are in vacation, statement 1-3 immediately hold. If  $Y_i^+(0) = Y_i^{+-}(0)$ , then  $Y_i^+(0) = 0$  and service station  $i$  is in the last case as in Lemma 4, hence both  $i$  and  $i^-$  are in vacation and statement 2 holds. In summary, Statement 1-3 all hold for time  $t(0) = 0$  and thus the result follows.  $\square$

**7.2. Recapitulation of the Main Procedure and Proof of Theorem 2.** In order to prove Theorem 2, we need to recapitulate on the execution of our Main Procedure. Let us go back to equation (6) and allow us write

$$Y_i^+(t; T, y) = Y_i^+(t; T)$$

to recognize the boundary condition in (6). Moreover, we recall that  $y_i = \bar{Y}_i'(-T)$ , from equation (7). For any  $T > 0$  define the event

$$C_T = \{\text{for all } t \in [0, T] \text{ there is } i \text{ such that } Y_i^+(k; T, \bar{Y}'(-T)) > 0\}.$$

Then put  $\bar{\tau} = \inf\{t \geq 0 : \bar{C}_t \text{ occurs}\}$ . Assuming that the output indeed follows the steady state distribution, the statement of Theorem 2 concerning the computational cost measure in terms of random numbers generated will follows if we can show that there exists  $\delta > 0$  such that  $E[\exp(\delta \bar{\tau})] < \infty$ .

We start by noting that

$$P(\bar{\tau} > u) = P(C_u).$$

In order to compute  $P(C_n)$  we can think forward in time, in particular consider

$$\begin{aligned} d\hat{Y}_i^+(t; 0, y) &= dN_{0,i}(t; 0) + \sum_{j: j \neq i, 1 \leq j \leq d} I(\hat{S}_j^+(t_-; 0, y) > 0) dD_{j,i}(t; 0) - I(\hat{Y}_i^+(t_-; 0, y) > 0) dD_i(t; 0), \\ d\hat{S}_i^+(t; 0, y) &= (I(\hat{Y}_i^+(t_-; 0, y) > 0) - I(\hat{S}_i^+(t_-; 0, y) > 0)) dD_i(t; 0), \\ \hat{Y}_i^+(0; 0, y) &= y, \quad \hat{S}_i^+(0; 0, y) = 1. \end{aligned}$$

Note the relation between  $D_i(t; 0)$  and  $D_i(-t)$ , defined in (8), in particular  $D_i(-t) = -D_i(t; 0) \leq 0$  (similarly  $N_{0,i}(-t) = N_{0,i}(t; 0)$ ). Then let  $Y_i^+(t; 0, y) = \hat{Y}_i^+(t; 0, y) + \hat{S}_i^+(t; 0, y)$  we have that

$$P(C_u) = P(\text{for all } t \in [0, u], \text{ there is } i \text{ such that } Y_i^+(t; 0, \bar{Y}'(0)) > 0).$$

The strategy is to first describe the evolution of  $Y^+(\cdot; 0, y)$  in terms of a Markov process. We need to track the residual times associated with each renewal process and the number of people both in queue and in service in each station. In particular, define

$$G_i(t) = \sup\{|A_i(-n)| : 1 \leq n \leq N_{0,i}(t; 0) + 1\} - t.$$

Similarly, we define

$$H_i(t) = \sup\{|B_i(-n)| : 1 \leq n \leq D_i(t; 0)\} - t.$$

Then we let  $t(n)$  be the times at which events occur, that is,  $t(1) < t(2) < \dots$  are the discontinuity points of the process  $\{E(t) : t \geq 0\}$  defined as  $E(t) = \sum_i N_{0,i}(t; 0) + \sum_i D_i(t; 0)$ . Let us write  $\Theta_i^+(n) = (\hat{Y}_i^+(t(n)), \hat{S}_i^+(t(n)))$  and define  $\Xi^+(n) = (\Xi_i(n) : 1 \leq i \leq d)$  as

$$\Xi_i^+(n) = (\Theta_i^+(n), G_i(t(n)), H_i(t(n))).$$

Note that  $\{\Xi^+(n) : n \geq 0\}$  forms a Markov chain and we are given the initial condition  $\Xi_i^+(0) = (\bar{Y}_i'(0), 1, G_i(0), H_i(0))$ . Now, define

$$\tau(c) = \inf\{n \geq 0 : \sum_{i=1}^d \Theta_i^+(n) \leq c\},$$

for some  $c > 0$ . Following a similar approach to [7], due to Assumption 2, we now can show that there exists  $c > 0$  such that  $E[\exp(\delta\tau(c))] < \infty$ . Moreover, because the inter-arrivals have unbounded support a geometric trial argument will yield that if  $\delta > 0$  is chosen sufficiently small then  $E[\exp(\delta\tau(0))] < \infty$ . In turn, this bound implies that  $E[\exp(\delta\bar{\tau})] < \infty$ .

Next we want to show that the output indeed follows the target steady state distribution. This portion follows precisely from the validity of the DCFTP protocol.

Using the similar notation of  $Y^+(t; T, y)$ , we define  $Y(t; T, y)$  as the number of customers in a GJN start with  $Y(0; T, y) = y$  and is driven by the same sequence of inter-arrival times, service requirements and routing indices as  $\mathcal{N}^+$  on  $[-T, 0]$ . Given the comparison results in Theorem 1, given that  $Y^+(-T'') = 0$ , we can conclude that for all  $T > T''$ ,

$$\sum_i Y_i(T - T''; T, 0) \leq \sum_i Y_i(T - T''^+(-T)) \leq \sum_i Y_i^+(T - T''^+(-T)) = \sum_i Y_i^+(-T'') = 0$$

and hence  $Y(T - T''; T, 0) = 0$ . Therefore, for any  $T > T''$

$$Y(T; T, 0) = Y(T; T, Y(T - T''; T, 0)) = Y(T''; T'', 0).$$

As the process  $Y(\cdot)$  has a unique stationary distribution (see [12]), we can conclude  $Y(T''; T'', 0) = \lim_{T \rightarrow \infty} Y(T; T, 0)$  follows the stationary distribution.

**Acknowledgement:** Blanchet acknowledges support from the NSF through the grants CMMI-0846816 and 1069064. Chen acknowledges support from the NSF through the grant CMMI-1538102.

## REFERENCES

- [1] Blanchet, J. and Chen, X. (2014). Steady-state simulation of reflected Brownian motion and related stochastic networks. *Annals of Applied Probability*, Vol. 25, pp 3209-3250.
- [2] Blanchet, J. and Sigman, K. (2011). On exact sampling of stochastic perpetuities. *Journal of Applied Probability*, Special Vol. 48A, pp 165-183.
- [3] Blanchet, J. and Wallwater, A. (2014). Exact sampling of stationary and time-reversed queues. *ACM Trans. Model. Comput. Sim.*, Vol. 25, Article 26.
- [4] Basic, A., Durand, S., Gaujal, B., Perronnin, F. (2015) Perfect sampling of Jackson queueing networks. *Queueing Systems*, Vol. 30, pp. 223-260.
- [5] Chang, C., Thomas, J. A. and Kiang, S. (1994). On the stability of open networks: a unified approach by stochastic dominance. *Queueing Systems*, Vol. 15, pp 239-260.
- [6] Ensor, K. B. and Glynn, P. W. (2000). Simulating the maximum of a random walk. *Journal of Statistical Planning and Inference*, Vol. 85, pp 127-135.
- [7] Gamarnik, D. and Zeevi, A. (2006). Validity of heavy traffic steady-state approximations in generalized Jackson networks. *Annals of Applied Probability*, Vol 16, pp 56-90.
- [8] Harrison, J. M. and Reiman, M. I. (1981). Reflected Brownian motion on an orthant. *Annals of Probability*, Vol. 9, pp 302-308
- [9] Kendall, W. (2004). Geometric ergodicity and perfect simulation. *Electronic Communications in Probability*, Vol. 9, pp 140-151.
- [10] Murdoch, D. J. and Takahara, G. (2006). Perfect sampling for queues and network models. *TOMACS*, Vol. 16, pp 76-92.
- [11] Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures & Algorithms*, Vol. 9, pp 223-252.
- [12] Sigman, K. (1990). The stability of open queueing networks. *Stochastic Processes and their Applications*, Vol. 35, pp 11-25.

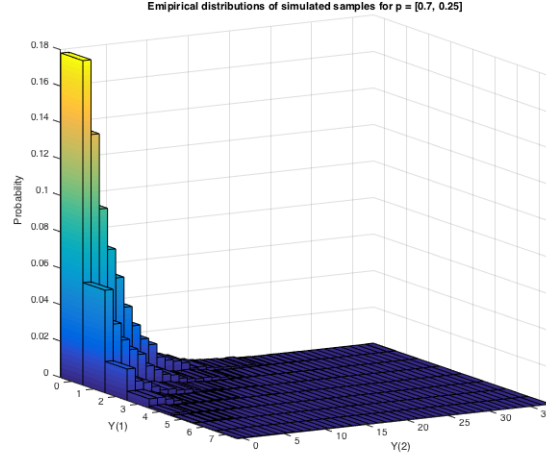
COLUMBIA UNIVERSITY, COLUMBIA UNIVERSITY, DEPARTMENT OF INDUSTRIAL ENGINEERING & OPERATIONS RESEARCH.

*E-mail address:* jose.blanchet@columbia.edu

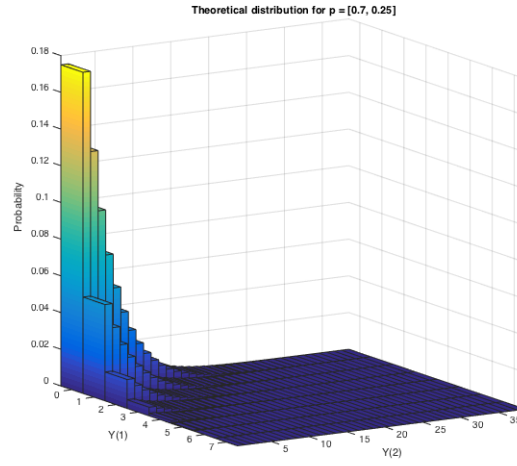
WUHAN UNIVERSITY, WUHAN UNIVERSITY, ECONOMICS AND MANAGEMENT SCHOOL

*E-mail address:* little4cat@gmail.com

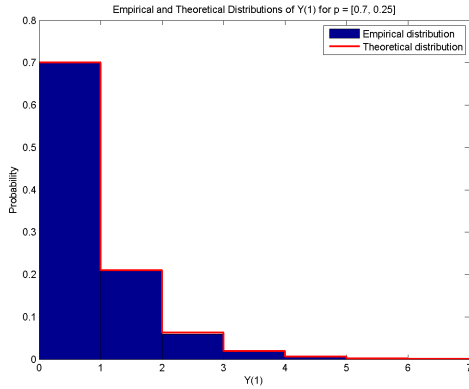




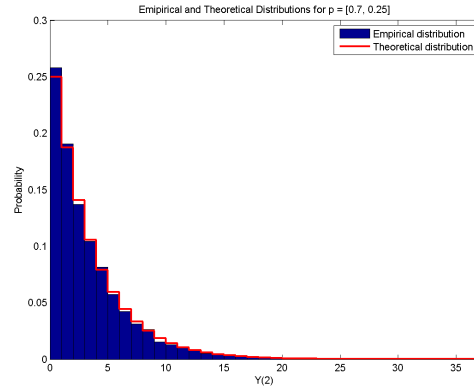
(A)



(B)

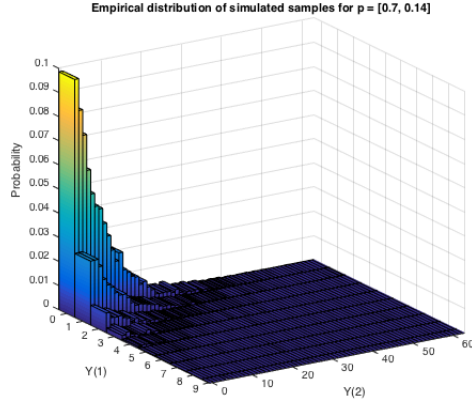


(C)

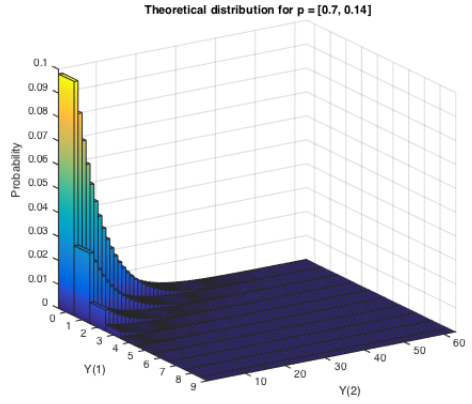


(D)

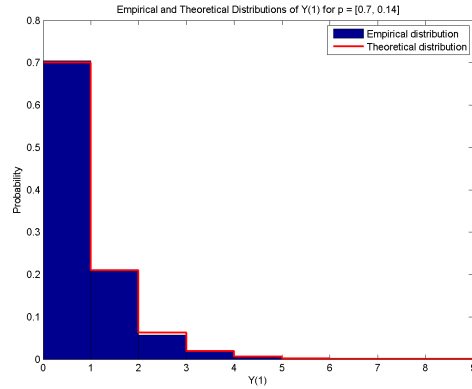
FIGURE 1.  $\lambda = (0.225, 0.717)$ ,  $\mu = (1, 1)$ . (A) Histogram of the 10000 simulated samples of  $(Y_1(\infty), Y_2(\infty))$ . (B) Theoretic steady-state distribution of  $(Y_1(\infty), Y_2(\infty))$ . (C) Marginal distribution of  $Y_1(\infty)$ . (D) Marginal distribution of  $Y_2(\infty)$ .



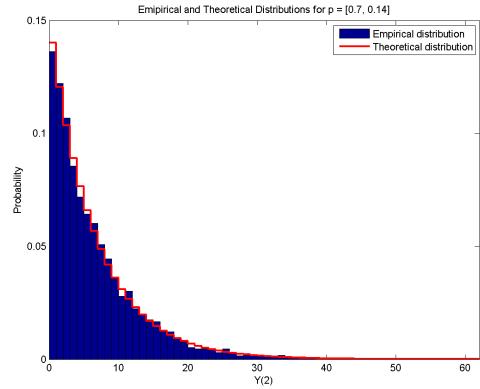
(A)



(B)



(C)



(D)

FIGURE 2.  $\lambda = (0.214, 0.827)$ ,  $\mu = (1, 1)$ . (A) Histogram of the 10000 simulated samples of  $(Y_1(\infty), Y_2(\infty))$ . (B) Theoretic steady-state distribution of  $(Y_1(\infty), Y_2(\infty))$ . (C) Marginal distribution of  $Y_1(\infty)$ . (D) Marginal distribution of  $Y_2(\infty)$ .